



TUGAS AKHIR - KI141502

IMPLEMENTASI *GESTALT PRINCIPLES* PADA RANCANG BANGUN APLIKASI BERBASIS ANDROID CLEARROUTE

AZKAA KHOIRUDDIN
NRP 5113100135

Dosen Pembimbing I
Dr. Tech. Ir. R. V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Wijayanti Nurul K, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

Implementasi Gestalt Principles pada Rancang Bangun Aplikasi berbasis Android ClearRoute

**Azkaa Khoiruddin
NRP 5113100135**

**Dosen Pembimbing I
Dr. Tech. Ir. R. V. Hari Ginardi, M.Sc.**

**Dosen Pembimbing II
Wijayanti Nurul K, S.Kom., M.Sc.**

**DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

Implementation of Gestalt Principles to Design ClearRoute Android Application

**Azkaa Khoiruddin
NRP 5113100135**

**Supervisor I
Dr. Tech. Ir. R. V. Hari Ginardi, M.Sc.**

**Supervisor II
Wijayanti Nurul K, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

IMPLEMENTASI GESTALT PRINCIPLES PADA RANCANG BANGUN APLIKASI BERBASIS ANDROID CLEARROUTE

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Algoritma dan Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

AZKAA KHOIRUDDIN
NRP : 5113100135

Disetujui oleh Pembimbing Tugas Akhir

1. Dr. Tech. Ir. R. V. Hari Gmardi, M.Sc.
NIP: 19650518 199203 1 003 (Pembimbing 1)
2. Wijayanti Nurul K, S.Kom, M.Sc.
NIP: 19860312 201212 2 004 (Pembimbing 2)

SURABAYA
JUNI, 2017

[Halaman ini sengaja dikosongkan]

IMPLEMENTASI GESTALT PRINCIPLES PADA RANCANG BANGUN APLIKASI BERBASIS ANDROID CLEARROUTE

Nama Mahasiswa : Azkaa Khoiruddin
NRP : 5113100135
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr. Tech. Ir. R. V. Hari Ginardi, M.Sc.
Dosen Pembimbing 2 : Wijayanti Nurul K, S.Kom, M.Sc.

Abstrak

Sebelum seseorang melakukan perjalanan akan ada beberapa hal yang perlu dipersiapkan. Baik itu berupa perlengkapan ataupun informasi mengenai kondisi rute perjalanan. Salah satu informasi yang penting untuk didapatkan adalah kondisi cuaca. Namun sekarang ini informasi mengenai kondisi cuaca ataupun ramalan cuaca bersifat umum untuk satu kota dan belum ada informasi cuaca berdasarkan lokasi yang spesifik dengan perjalanan seseorang untuk mengetahui kondisi cuaca lokal yang akan dilaluinya. Untuk itu diperlukan sebuah aplikasi yang dapat memudahkan seseorang dalam menemukan informasi cuaca yang spesifik dengan lokasi atau rute perjalanan yang akan ditempuh.

Aplikasi Clearroute adalah aplikasi yang dibangun untuk memudahkan seseorang mengetahui informasi cuaca pada rute perjalanan yang akan dilaluinya. Aplikasi ini dibangun menggunakan platform perangkat bergerak yang memiliki fungsi-fungsi untuk memudahkan seseorang mencari informasi cuaca untuk perjalanannya.

Pada aplikasi perangkat bergerak tersebut, diperlukan sebuah desain sistem yang mudah dan nyaman digunakan oleh pengguna. Dalam ilmu hubungan interaksi manusia dan komputer terdapat beberapa prinsip yang dapat digunakan

sehingga sebuah aplikasi dapat digunakan secara ideal oleh masyarakat luas.

Pada aplikasi ini memanfaatkan tiga dari enam prinsip yang difokuskan pada desain antarmukanya. Tiga prinsip itu ialah figure/ground, proximity, dan common fate. Kemudian sistem ini menggunakan Native Framework sebagai kerangka kerja pemrograman.

Pengujian pada sistem ini dilakukan dengan cara Black Box yaitu pengujian fungsionalitas aplikasi oleh pengguna dengan karakteristik pengendara di wilayah Surabaya. Selain itu, juga menggunakan heuristic evaluation kepada pengguna ahli dalam bidang desain antarmuka. Dari hasil pengujian, sistem yang telah dirancang dan diimplementasikan telah memenuhi segala kebutuhan dan kenyamanan pengguna pada aplikasi Clearroute.

Kata kunci: Clearroute, User Interface, Gestalt Principles, User Experience, Sosial, Cuaca

IMPLEMENTATION OF GESTALT PRINCIPLES TO DESIGN CLEARROUTE ANDROID APPLICATION

Student's Name : Azkaa Khoiruddin
Student's ID : 5113100135
Department : Teknik Informatika FTIF-ITS
First Advisor : Dr. Tech. Ir. R. V. Hari Ginardi,
M.Sc.
Second Advisor : Wijayanti Nurul K, S.Kom, M.Sc.

Abstract

There are some things that need to be prepared before someone travels, whether it is in the form of equipment or information about the travel route condition. One out of important informations to have before travelling is the weather condition. But, the available information about wheather condition or its forecast now is only in general for a city and there is no weather information for a spesific location according to the route of the trip to find out the weather conditions to go through. Therefore, an application that is able to help someone in finding spesific weather informations according to the location or travel route to be taken is required.

Clearroute application is an application that is built in order to facilitate people to get the weather information for a spesific route that is going to be taken. This application is built for mobile device platforms that has the functions of making it easier for people to search about weather information for their trip.

In this mobile device application, an easy and comfortable design system is needed by users. There are some principles in Human-Computer Interaction that can be used by an application to be look like ideal application.

This application utilize three of six principles in its interfaces. Those principles is proximity, figure/ground, and

common fate. This application make use of Native Framework as programming framework.

There are two methods to test this application. Blackbox is a method to test its functionality. Blackbox is using participants with unique characteristics such as he/she can drives a vehicle in Surabaya. The other method is heuristic evaluation. Heuristic evaluation is using expert user to test its design interfaces. The result says that this application fulfill every needs and cosiness to be used by users.

Keywords: ClearRoute, User Interface, Gestalt Principle, User Experience, Social, Weather

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul :

“Implementasi Gestalt Principles pada Rancang Bangun Aplikasi berbasis Android Clearroute”

Harapan dari penulis semoga apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan tugas akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Zaenal Abidin, Ibu Mariati, Rif’atul Wafirotn Nisa’, dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan tugas akhir ini.
2. Bapak Hari Ginardi dan Ibu Wijayanti Nurul Khotimah selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan Jurusan Teknik Informatika ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Sahabat-sahabat M Fikri Alauddin, Abednego Indra A, Rifki Dwisetyo W, Angga Setyawan, Ziyad Syauqi F, Demsey Iman M, Nida Amalia yang sudah banyak

merasakan pahit manis kehidupan bersama penulis di masa pendidikan.

6. Kekasih, Puji Mar'atus Sholikhah yang sudah banyak memberikan pelajaran kehidupan dan membantu penulis untuk bangkit ketika terpuruk di masa studinya.
7. Teman-teman Kabinet BPH SCHEMATICS HMTC 2015, GenBI ITS 2016 yang telah memberikan pengalaman berharga kepada penulis selama masa studi penulis.
8. Teman-teman penghuni Workshop Pemrograman (LP), Rina Wijaya KW, Nafiar Rachmansyah, Hari Setyawan, M Fahrul Razi, Ronauli Silva NS, Afiiif Naufal, Ardi Nusawan, Dewangga Winardi, Bayu Sektiaji, Nobby Phala yang telah memberikan banyak dukungan dan semangat kepada penulis.
9. Teman-teman angkatan 2013 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama 4 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan tugas akhir ini.
10. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan tugas akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2017

Azkaa Khoiruddin

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
Abstrak	ix
Abstract	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR KODE SUMBER	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 <i>Android Application</i>	7
2.2 Badan Meteorologi Klimatologi dan Geofisika (BMKG)	9
2.3 <i>Location Based Service (LBS)</i>	11
2.4 <i>Human-Computer Interaction</i>	12
2.4.1 <i>Gestalt Principles</i>	13
2.4.2 <i>Heuristic Evaluation</i>	19
BAB III ANALISIS DAN PERANCANGAN.....	21
3.1 Deskripsi Umum.....	21
3.2 Arsitektur Sistem.....	21
3.2.1 <i>Applications</i>	23
3.2.2 <i>Backend Services</i>	24
3.2.3 <i>Internal Data Resources</i>	25
3.2.4 <i>External Data Resource</i>	27
3.3 Analisis Proses Bisnis.....	29

3.4 Analisis Kebutuhan Sistem.....	30
3.5 Analisis Aktor.....	31
3.6 Perancangan Sistem.....	32
3.6.1 Kasus Penggunaan.....	32
3.6.2 Antarmuka Aplikasi.....	48
BAB IV IMPLEMENTASI.....	57
4.1 Lingkungan Implemetasi	57
4.2 Implementasi Perangkat Lunak	58
4.2.1 Implementasi Kasus Penggunaan	58
4.2.2 Implementasi Antarmuka	67
BAB V UJI COBA DAN EVALUASI	89
5.1 Lingkungan Uji Coba	89
5.2 Skenario Uji Coba.....	89
5.2.1 Pengujian <i>Heuristic</i>	90
5.2.2 Pengujian Fungsionalitas.....	91
5.3 Evaluasi Pengujian.....	105
5.3.1 Evaluasi Pengujian <i>Heuristic</i>	105
5.3.2 Evaluasi Pengujian Fungsionalitas	120
BAB VI KESIMPULAN DAN SARAN.....	125
6.1 Kesimpulan.....	125
6.2 Saran	126
DAFTAR PUSTAKA.....	127
LAMPIRAN	129
8.1 Kode Sumber pada Aplikasi	129
8.2 Hasil Pengujian dan Kuisisioner	136
BIODATA PENULIS.....	147

DAFTAR KODE SUMBER

Kode Sumber 5.1 Revisi Masalah 3	111
Kode Sumber 8.1 Fungsi Dexter	129
Kode Sumber 8.2 Pengaturan Proximity pada Halaman Login	130
Kode Sumber 8.3 Fungsi Login	130
Kode Sumber 8.4 Fungsi Pengaturan Animasi pada Grafik	131
Kode Sumber 8.5 Pengaturan Proximity pada Halaman Signup	132
Kode Sumber 8.6 Pengaturan Common Fate dan Proximity pada Halaman Ramalan.....	133
Kode Sumber 8.7 Fungsi SlidingPanel	134
Kode Sumber 8.8 Fungsi TabLayout	134
Kode Sumber 8.9 Fungsi ToolbarMenu	134
Kode Sumber 8.10 Fungsi Update Profil	135

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Antarmuka Aplikasi Waze	8
Gambar 2.2 Antarmuka Aplikasi Google Maps	8
Gambar 2.3 Antarmuka Aplikasi Info BMKG	9
Gambar 2.4 Kotak Dialog <i>Distribution List Membership</i> Pada Outlook.....	14
Gambar 2.5 Discreet's Software Installer	14
Gambar 2.6 Mac OS <i>Page Setup Dialog Box</i>	15
Gambar 2.7 Mac OS Dan Component One.....	16
Gambar 2.8 <i>Icon</i> Dokumen	16
Gambar 2.9 <i>Pop Up</i> Pada Graceusa.org.....	17
Gambar 2.10 Gap-Minder Menganimasikan Titik-Titik Untuk Menunjukkan Negara-Negara Yang Mengalami Perkembangan Yang Mirip	18
Gambar 2.11 <i>Drag Folder</i> Bersama.....	19
Gambar 3.1 Diagram Arsitektur Aplikasi ClearRoute	22
Gambar 3.2 Diagram Komponen <i>Backend-Services</i>	25
Gambar 3.3 Desain PDM Basis Data	26
Gambar 3.4 Data Citra Ramalan Cuaca BMKG	28
Gambar 3.5 Keterangan Kondisi Cuaca	28
Gambar 3.6 Diagram Alir Proses Bisnis	30
Gambar 3.7 Diagram Kasus Penggunaan.....	32
Gambar 3.8 Diagram Aktivitas Mendaftar Akun.....	35
Gambar 3.9 Diagram Aktivitas Melakukan <i>Login</i>	37
Gambar 3.10 Diagram Aktivitas Melihat Profil	38
Gambar 3.11 Diagram Aktivitas Mengubah Profil	40
Gambar 3.12 Diagram Aktivitas Melihat Ramalan Terkini ..	42
Gambar 3.13 Diagram Aktivitas Melihat Berita Cuaca	43
Gambar 3.14 Diagram Aktivitas Membuat Berita Cuaca	45
Gambar 3.15 Diagram Aktivitas Melihat Rekomendasi Rute	47
Gambar 3.16 Diagram Aktivitas Melihat Grafik.....	48
Gambar 3.17 Antarmuka Halaman <i>Register</i>	49

Gambar 3.18 Antarmuka Halaman <i>Login</i>	50
Gambar 3.19 Antarmuka Halaman Cuaca.....	51
Gambar 3.20 Antarmuka Halaman Berita	52
Gambar 3.21 Antarmuka Halaman Peta.....	53
Gambar 3.22 Antarmuka Halaman Grafik	54
Gambar 3.23 Antarmuka Halaman Profil.....	55
Gambar 3.24 Antarmuka Halaman Tambah Berita	56
Gambar 4.1 <i>Pseudocode</i> Mendaftar Akun Pada Aplikasi	59
Gambar 4.2 <i>Pseudocode</i> Mendaftar Akun Pada <i>Web Service</i>	59
Gambar 4.3 <i>Pseudocode</i> Melakukan <i>Login</i> pada Aplikasi....	60
Gambar 4.4 <i>Pseudocode</i> Melakukan <i>Login</i> pada <i>Web Service</i>	60
Gambar 4.5 <i>Pseudocode</i> Melihat Profil pada Aplikasi	61
Gambar 4.6 <i>Pseudocode</i> Melihat Profil pada <i>Web Service</i> ...	62
Gambar 4.7 <i>Pseudocode</i> Mengubah Profil pada Aplikasi.....	63
Gambar 4.8 <i>Pseudocode</i> Mengubah Profil pada <i>Web Service</i>	63
Gambar 4.9 <i>Pseudocode</i> Melihat Ramalan pada Aplikasi	63
Gambar 4.10 <i>Pseudocode</i> Melihat Berita pada Aplikasi.....	64
Gambar 4.11 <i>Pseudocode</i> Melihat Berita pada <i>Web Service</i>	64
Gambar 4.12 <i>Pseudocode</i> Membuat Berita pada Aplikasi	65
Gambar 4.13 <i>Pseudocode</i> Membuat Berita pada <i>Web Service</i>	65
Gambar 4.14 <i>Pseudocode</i> Melihat Rute pada Aplikasi.....	66
Gambar 4.15 <i>Pseudocode</i> Melihat Grafik pada Aplikasi	67
Gambar 4.16 Antarmuka Halaman <i>Splash Screen</i>	68
Gambar 4.17 Antarmuka Halaman <i>Login</i>	69
Gambar 4.18 <i>Gestalt Principles</i> Pada Halaman <i>Login</i>	69
Gambar 4.19 Antarmuka Halaman <i>Register</i>	70
Gambar 4.20 <i>Gestalt Principles</i> Pada Halaman <i>Register</i>	71
Gambar 4.21 Antarmuka Halaman <i>Forgot Password</i>	73
Gambar 4.22 <i>Gestalt Principles</i> Pada Halaman <i>Forgot Password</i>	74
Gambar 4.23 Antarmuka Halaman Ramalan.....	75

Gambar 4.24 <i>Gestalt Principles</i> Pada Halaman Ramalan.....	76
Gambar 4.25 Antarmuka Halaman Berita.....	78
Gambar 4.26 <i>Gestalt Principles</i> Pada Halaman Berita	79
Gambar 4.27 Antarmuka Halaman Tambah Berita.....	80
Gambar 4.28 <i>Gestalt Principles</i> Pada Halaman Tambah Berita	81
Gambar 4.29 Antarmuka Halaman Peta.....	82
Gambar 4.30 <i>Gestalt Principles</i> Pada Halaman Peta	83
Gambar 4.31 Antarmuka Halaman Grafik	84
Gambar 4.32 <i>Gestalt Principles</i> Pada Halaman Grafik	85
Gambar 4.33 Antarmuka Halaman Profil.....	86
Gambar 4.34 <i>Gestalt Principles</i> Pada Halaman Profil.....	87
Gambar 5.1 Hasil Pengujian Mendaftar Akun	93
Gambar 5.2 Hasil Pengujian Melakukan <i>Login</i>	94
Gambar 5.3 Hasil Pengujian Melihat Profil	96
Gambar 5.4 Hasil Pengujian Mengubah Profil	97
Gambar 5.5 Hasil Pengujian Melihat Ramalan Terkini	99
Gambar 5.6 Hasil Pengujian Melihat Berita Cuaca	100
Gambar 5.7 Hasil Pengujian Membuat Berita Cuaca.....	102
Gambar 5.8 Hasil Pengujian Melihat Rekomendasi Rute...	103
Gambar 5.9 Hasil Pengujian Melihat Grafik.....	105
Gambar 5.10 Bukti Masalah 1.....	107
Gambar 5.11 Revisi Masalah 1	108
Gambar 5.12 Bukti Masalah 2.....	109
Gambar 5.13 Revisi Masalah 2	110
Gambar 5.14 Bukti Masalah 3.....	111
Gambar 5.15 Bukti Masalah 4.....	112
Gambar 5.16 Revisi Masalah 4	113
Gambar 5.17 Bukti Masalah 5.....	114
Gambar 5.18 Revisi Masalah 5	115
Gambar 5.19 Bukti Masalah 6.....	116
Gambar 5.20 Revisi Masalah 6	116
Gambar 5.21 Bukti Masalah 7.....	117
Gambar 5.22 Revisi Masalah 7	118
Gambar 5.23 Bukti Masalah 8.....	119

Gambar 5.24 Revisi Masalah 8	120
Gambar 8.1 Objek Uji Coba 1	136
Gambar 8.2 Objek Uji Coba 2	137
Gambar 8.3 Objek Uji Coba 3	138
Gambar 8.4 Objek Uji Coba 4	139
Gambar 8.5 Objek Uji Coba 5	140
Gambar 8.6 Objek Uji Coba 6	141
Gambar 8.7 Objek Uji Coba 7	142
Gambar 8.8 Objek Uji Coba 8	143
Gambar 8.9 Objek Uji Coba 9	144
Gambar 8.10 Objek Uji Coba 10	145
Gambar 8.11 Hasil Evaluasi <i>Heuristic</i>	146

DAFTAR TABEL

Tabel 3.1 Daftar Kebutuhan Fungsional Sistem	31
Tabel 3.2 Keterangan Kode Kasus Penggunaan	33
Tabel 3.3 Spesifikasi Kasus Penggunaan Mendaftar Akun ..	33
Tabel 3.4 Spesifikasi Kasus Penggunaan Melakukan Login.	35
Tabel 3.5 Spesifikasi Kasus Penggunaan Melihat Profil	37
Tabel 3.6 Spesifikasi Kasus Penggunaan Mengubah Profil..	39
Tabel 3.7 Spesifikasi Kasus Penggunaan Melihat Ramalan Terkini	40
Tabel 3.8 Spesifikasi Kasus Penggunaan Melihat Berita Cuaca	42
Tabel 3.9 Spesifikasi Kasus Penggunaan Membuat Berita Cuaca.....	44
Tabel 3.10 Spesifikasi Kasus Penggunaan Melihat Rekomendasi Rute	45
Tabel 3.11 Spesifikasi Kasus Penggunaan Melihat Grafik ...	47
Tabel 5.1 Spesifikasi Lingkungan Pengujian	89
Tabel 5.2 Poin Heuristic	90
Tabel 5.3 Tingkat Gangguan	91
Tabel 5.4 Tingkat Perbaikan	91
Tabel 5.5 Skenario Pengujian Mendaftar Akun	92
Tabel 5.6 Skenario Pengujian Melakukan Login	93
Tabel 5.7 Skenario Pengujian Melihat Profil	95
Tabel 5.8 Skenario Pengujian Mengubah Profil	96
Tabel 5.9 Skenario Pengujian Melihat Ramalan Terkini	98
Tabel 5.10 Skenario Pengujian Melihat Berita Cuaca	99
Tabel 5.11 Skenario Pengujian Membuat Berita Cuaca.....	101
Tabel 5.12 Skenario Pengujian Melihat Rekomendasi Rute	102
Tabel 5.13 Skenario Pengujian Melihat Grafik.....	104
Tabel 5.14 Daftar Masalah Dari Pengujian	106
Tabel 5.15 Analisa Heuristic Permasalahan.....	106
Tabel 5.16 Hasil Pengujian Fungsional.....	120
Tabel 5.17 Waktu Pengujian Fungsional	121

Tabel 5.18 Hasil Pengujian Antarmuka.....	122
---	-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap orang pasti pernah melakukan perjalanan. Dalam setiap perjalanan tentu banyak pertimbangan yang akan diperhitungkan untuk melaksanakan perjalanan tersebut. Beberapa diantaranya adalah rute yang dilalui dan cuaca. Rute merupakan alur dan arah yang akan dilalui untuk mencapai tempat tujuan, dan cuaca adalah keadaan udara pada saat tertentu dan di wilayah tertentu yang relatif sempit (tidak luas) dan pada jangka waktu yang singkat.

Di Indonesia, terdapat Badan yang sudah secara resmi memiliki tugas untuk mengawasi dan memberikan berita mengenai cuaca yang sedang berlangsung dan akan terjadi, yaitu BMKG (Badan Meteorologi Klimatologi dan Geofisika). BMKG secara *realtime* memperbaharui berita cuaca yang ada, agar dapat dilihat oleh masyarakat umum. BMKG Surabaya menggunakan acuan beberapa Pos Hujan yang biasanya merupakan kantor desa atau tempat umum besar lainnya untuk memberikan informasi mengenai cuaca di sekitar Pos Hujan tersebut.

Pengembangan aplikasi *mobile* saat ini telah mengalami perkembangan yang cukup pesat. Baik dari segi sistem maupun *user interfaces*. Penggunaan desain aplikasi sangat menentukan pengguna dalam menjalankan fungsionalitas aplikasi tersebut. Desain aplikasi bukan hanya dari segi seni aplikasi melainkan juga dari segi kemudahan penggunaan fungsi atau fitur di dalamnya. Seringkali pengguna merasa bingung atau ambigu terhadap aplikasi dengan desain yang buruk. Oleh karena itu, dibutuhkan *user interface* yang dapat memudahkan pengguna dalam menjalankan fungsionalitas aplikasi secara penuh.

Dengan menggunakan prinsip *Gestalt* pada desain antarmuka aplikasi, diyakini hal tersebut akan membuat masyarakat menjadi lebih mudah menggunakan aplikasinya.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana membangun aplikasi (ClearRoute) penentuan rute jalan berdasarkan kondisi cuaca berbasis Android dengan dukungan data dari BMKG?
2. Bagaimana mengimplementasikan *Gestalt Principles* pada setiap desain aplikasi ClearRoute?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. *Gestalt Principles* yang digunakan ialah *Proximity*, *Figure/Ground*, dan *Common Fate*.
2. Data peta yang digunakan hanya wilayah Surabaya.
3. Menggunakan *native framework* bawaan dari Android Studio dengan bahasa pemrograman utama yaitu Java.
4. Basis data yang digunakan dalam aplikasi yaitu MySQL.
5. Desain *icon* maupun *image* menggunakan Google Material Icon atau Font Awesome Icon.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Membangun aplikasi android penentuan rute jalan berdasarkan kondisi cuaca dengan nama aplikasi yaitu ClearRoute.
2. Memberikan data cuaca akurat kepada pengguna dengan sumber data resmi BMKG.
3. Mengimplementasikan *Gestalt Principles* dalam pembangunan aplikasi ClearRoute khususnya pada desain antarmukanya.

1.5 Manfaat

Manfaat yang diperoleh dari pembuatan Tugas Akhir ini adalah dapat membantu masyarakat mempermudah penggunaan aplikasi yang mana aplikasi tersebut dapat mengetahui kondisi cuaca pada jalur perjalanan yang akan dilaluinya. Sehingga pengguna dapat mempersiapkan segala kebutuhan perjalanannya sesuai dengan kondisi cuaca yang akan dilalui.

1.6 Metodologi

Ada beberapa tahap dalam proses pengerjaan Tugas Akhir ini, mulai dari pencarian literatur, pengimplementasian, uji coba, hingga pembuatan laporan akhir. Berikut adalah tahap-tahap yang dilakukan dalam pembuatan Tugas Akhir ini.

a. Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi menggunakan literatur *paper* yang berasal dari jurnal internasional yaitu IEEE, *e-book* yang diberikan oleh dosen pembimbing untuk mencari informasi dan teori-teori yang berhubungan dengan *Gestalt Principles* yang dapat digunakan untuk mengerjakan Tugas Akhir ini. Kemudian digunakan juga pustaka pendukung lainnya yaitu informasi mengenai sumber data cuaca yang berasal dari BMKG. Selain itu juga dilakukan perumusan awal mengenai fungsionalitas aplikasi yang akan diimplementasikan pada sistem yang dibuat.

b. Perancangan dan Desain Sistem

Pada tahap ini dilakukan perancangan sistem dengan menggunakan studi literatur dan mempelajari konsep aplikasi yang akan dibuat. Dengan bekal teori, metode, dan informasi yang sudah terkumpul pada tahap sebelumnya diharapkan dapat membantu dalam proses perancangan sistem. Tahap ini merupakan tahap terpenting pada bentuk awal atau *prototype* yang akan diimplementasikan kedepannya.

c. Implementasi

Pada tahap ini dilakukan tahap untuk membangun metode-metode yang telah diajukan pada proposal Tugas Akhir. Implementasi Tugas Akhir ini menggunakan bahasa pemrograman Java dengan kerangka kerja *Native Framework* dan basis data yang digunakan ialah basis data spasial dari MySQL.

d. Uji Coba dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap sistem yang dibangun. Pengujian dan evaluasi sistem dilakukan untuk mencari masalah yang mungkin timbul dan melakukan perbaikan jika ditemukan kesalahan pada sistem.

e. Penyusunan Laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini disusun dengan sistematika penulisan yang akan dijelaskan sebagai berikut.

BAB I. PENDAHULUAN

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

BAB II. TINJAUAN PUSTAKA

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

BAB III. ANALISIS DAN PERANCANGAN

Bab ini berisi pembahasan mengenai perancangan dari *Gestalt Principles* dan penjelasan mengenai fungsionalitas yang akan digunakan serta alur kerja dari sistem yang dibangun untuk Tugas Akhir ini.

BAB IV. IMPLEMENTASI

Bab ini menjelaskan implementasi yang berbentuk kode sumber dari pembangunan aplikasi, dan *Gestalt Principles* sebagai acuan pembuatan *layout* dan animasi.

BAB V. UJI COBA DAN EVALUASI

Bab ini berisikan hasil uji coba dari *Gestalt Principles* untuk memberikan desain yang mudah bagi pengguna dan diakui secara internasional yang sudah diimplementasikan pada kode sumber.

BAB VI. KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

Pada bab ini dijelaskan pustaka penunjang yang berkaitan dengan pembuatan Tugas Akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap aplikasi dan berguna sebagai penunjang dalam pengembangan sistem.

2.1 *Android Application*

Android adalah *software stack* untuk perangkat *mobile* yang mencakup sistem operasi dan *middleware*. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak[1]. SDK Android menyediakan alat dan API yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. SDK Android merupakan alat kompilasi kode bersama dengan data dan sumber daya *file* ke dalam paket Android dan *file* arsip dengan akhiran APK. Semua kode dalam *file* APK tunggal dianggap satu aplikasi. *File* APK yang sudah diberikan *Sign* bisa diunggah kedalam PlayStore agar bisa digunakan banyak orang dengan cakupan wilayah yang luas.

a. Waze

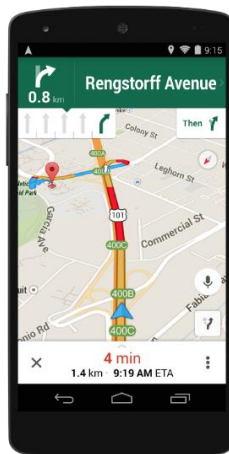
Waze adalah aplikasi komunitas berdasarkan kemacetan dan sebuah aplikasi navigasi bagi pengguna. Waze sangatlah populer digunakan di kalangan anak muda dikarenakan akurasi dalam memetakan kemacetan[2].



Gambar 2.1 Antarmuka Aplikasi Waze

b. Google Maps

Google Maps adalah aplikasi yang dapat melakukan navigasi untuk pengguna dalam sebuah perjalanan. Aplikasi yang dirilis oleh Google ini sangat banyak penggunaannya meskipun dari segi akurasi kemacetan dianggap oleh beberapa kalangan tidak lebih baik dari Waze[3].



Gambar 2.2 Antarmuka Aplikasi Google Maps

c. Info BMKG

Info BMKG adalah aplikasi yang diluncurkan oleh BMKG Indonesia. Semua informasi mengenai prakiraan cuaca, iklim, kualitas udara, dan gempa bumi yang terjadi berbagai wilayah di Indonesia tercakup dalam satu aplikasi *mobile*[4].



Gambar 2.3 Antarmuka Aplikasi Info BMKG

2.2 Badan Meteorologi Klimatologi dan Geofisika (BMKG)

BMKG merupakan suatu lembaga pemerintah non departemen yang memiliki tugas untuk melaksanakan tugas pemerintahan di bidang meteorologi, klimatologi dan geofisika. Dalam melaksanakan tugasnya BMKG menyelenggarakan fungsi berupa:

- Perumusan kebijakan nasional dan kebijakan umum di bidang meteorologi, klimatologi, dan geofisika.
- Perumusan kebijakan teknis di bidang meteorologi, klimatologi, dan geofisika.
- Koordinasi kebijakan, perencanaan dan program di bidang meteorologi, klimatologi, dan geofisika.
- Pelaksanaan, pembinaan dan pengendalian observasi, dan pengolahan data dan informasi di bidang meteorologi, klimatologi, dan geofisika.
- Pelayanan data dan informasi di bidang meteorologi, klimatologi, dan geofisika.
- Penyampaian informasi kepada instansi dan pihak terkait serta masyarakat berkenaan dengan perubahan iklim.
- Penyampaian informasi dan peringatan dini kepada instansi dan pihak terkait serta masyarakat berkenaan dengan bencana karena faktor meteorologi, klimatologi, dan geofisika.
- Pelaksanaan kerja sama internasional di bidang meteorologi, klimatologi, dan geofisika.
- Pelaksanaan penelitian, pengkajian, dan pengembangan di bidang meteorologi, klimatologi, dan geofisika.
- Pelaksanaan, pembinaan, dan pengendalian instrumentasi, kalibrasi, dan jaringan komunikasi di bidang meteorologi, klimatologi, dan geofisika.
- Koordinasi dan kerja sama instrumentasi, kalibrasi, dan jaringan komunikasi di bidang meteorologi, klimatologi, dan geofisika.
- Pelaksanaan pendidikan dan pelatihan keahlian dan manajemen pemerintahan di bidang meteorologi, klimatologi, dan geofisika.
- Pelaksanaan pendidikan profesional di bidang meteorologi, klimatologi, dan geofisika.
- Pelaksanaan manajemen data di bidang meteorologi, klimatologi, dan geofisika.

- Pembinaan dan koordinasi pelaksanaan tugas administrasi di lingkungan BMKG.
- Pengelolaan barang milik/kekayaan negara yang menjadi tanggung jawab BMKG.
- Pengawasan atas pelaksanaan tugas di lingkungan BMKG[5].

2.3 *Location Based Service (LBS)*

Location Based Service (LBS) adalah suatu layanan yang menyediakan informasi dengan memanfaatkan informasi-informasi geografis yang ada pada suatu perangkat bergerak. Dalam menyediakan informasi, LBS biasanya didukung oleh peta yang ditampilkan dalam perangkat bergerak[6]. Layanan ini memungkinkan untuk melakukan perhitungan yang berhubungan dengan lokasi dan jarak seseorang, misalnya rute terdekat, tempat terdekat, dan lain-lain. LBS terdiri dari beberapa komponen, yaitu:

a. **Perangkat *Mobile***

Perangkat *mobile* digunakan untuk meminta informasi yang dibutuhkan pengguna, seperti teks, gambar, dan suara. Contoh perangkat *mobile* adalah laptop, *handphone*, dan lain-lain.

b. **Komponen Pelacakan**

Komponen pelacakan dalam hal ini adalah sistem yang digunakan untuk menentukan posisi suatu objek. Komponen pelacakan yang paling banyak digunakan terutama pada *smartphone* adalah *Global Positioning System (GPS)*. GPS bekerja dengan menggunakan satelit GPS yang memancarkan sinyal ke bumi lalu diterima oleh *smartphone*. Selain GPS, komponen pelacakan yang lain adalah *Wireless Positioning System* yang menggunakan sinyal *WiFi* untuk menentukan lokasi suatu objek. Pada pengerjaan Tugas Akhir ini, komponen pelacakan yang digunakan adalah *Global Positioning System (GPS)*.

c. Jaringan Komunikasi

Jaringan komunikasi digunakan untuk mengirimkan data pengguna dan permintaan layanan dari perangkat *mobile* ke penyedia layanan dan meminta informasi kembali untuk pengguna.

d. Penyedia Aplikasi dan Layanan

Penyedia layanan menawarkan sejumlah layanan yang berbeda kepada pengguna dan bertanggung jawab untuk pengolahan permintaan layanan. Layanan tersebut dapat berupa perhitungan posisi dan menemukan rute yang berhubungan dengan posisinya, atau bahkan mencari informasi yang spesifik tentang posisi objek yang lain.

e. Penyedia Konten dan Data

Penyedia layanan biasanya disimpan dan dikelola oleh pihak yang memberikan layanan pengelolaan data.

2.4 Human-Computer Interaction

Analisis aplikasi sangat penting dilakukan untuk menghasilkan sistem yang mudah, efektif, efisien, dan tepat guna bagi pengguna. Model interaksi antara manusia dengan sistem melibatkan lima komponen yaitu pengguna atau manusia, interaksi, sistem komputer, aktivitas dan lingkungan kerja. Kunci utama dari HCI adalah daya guna (*usability*). *Usability* adalah tingkat produk dapat digunakan yang ditetapkan oleh pengguna, untuk mencapai tujuan secara efektif, efisien dan memuaskan dalam menggunakannya (ISO, 1998). Nielsen menguatkan pengertian *usability* tersebut dengan mengatakan bahwa *usability* merupakan suatu atribut kualitas yang menilai kemudahan penggunaan antar muka, yang memungkinkan pengguna untuk menyelesaikan tugasnya dengan jelas, transparan, lincah dan berguna. Hal-hal yang berkaitan dengan interaksi diantaranya adalah antar muka (*user interface*), bahasa, orientasi pada alat dan perangkat (*tools and device*), fleksibilitas, kompatibilitas sistem dengan sistem lain dan komunikasi.

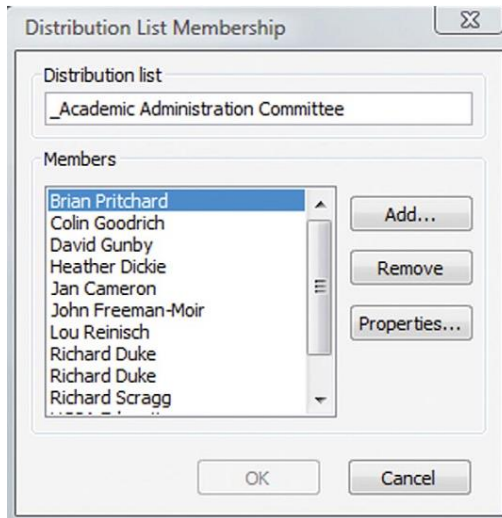
Salah satu bahasan terpenting dalam bidang HCI adalah antar muka pengguna (*user interface*), yang merupakan bagian sistem yang dikendalikan oleh pengguna untuk mencapai dan melaksanakan fungsi-fungsi suatu sistem. Pengguna berhubungan dengan sistem melalui antar muka pengguna. Peran antar muka pengguna dalam daya guna (*usability*) suatu sistem sangatlah penting. Oleh karenanya bentuk dan pembangunan antar muka pengguna perlu dilihat sebagai salah satu proses utama dalam keseluruhan pembangunan suatu sistem[7].

2.4.1 Gestalt Principles

Gestalt adalah sebuah teori yang menjelaskan proses persepsi melalui pengorganisasian komponen-komponen sensasi yang memiliki hubungan, pola atau kemiripan menjadi kesatuan[8]. Teori *Gestalt* berposisi terhadap teori strukturalisme. Teori *Gestalt* cenderung berupaya mengurangi pembagian sensasi menjadi bagian-bagian kecil. Teori *Gestalt* banyak dipakai dalam proses desain dan cabang seni rupa lainnya, karena banyak menjelaskan bagaimana persepsi visual bisa terbentuk. Persepsi jenis ini bisa terbentuk karena:

- *Proximity*

Proximity adalah jarak relatif antar objek yang mempengaruhi persepsi kita tentang bagaimana objek diatur dalam sub grup. Objek yang dekat satu sama lain tampak berada dalam satu grup dan objek yang berjauhan tidak dalam satu grup.



Gambar 2.4 Kotak Dialog *Distribution List Membership* Pada Outlook

Pada Gambar 2.4 terlihat *List button* (“Add”, “Remove”, “Properties”) berada dalam satu grup terpisah dengan window *button* (“Ok”, “Cancel”).

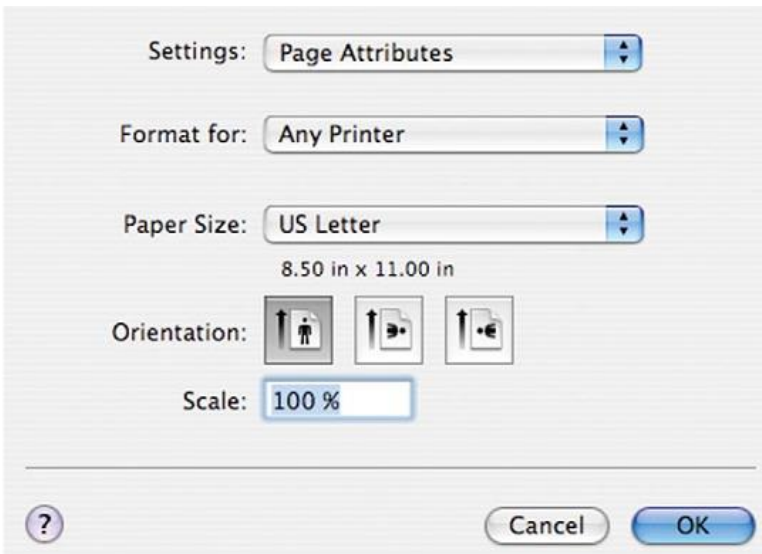


Gambar 2.5 Discreet’s Software Installer

Pada Gambar 2.5 terlihat *radio button* dikelompokkan secara vertikal. Hal ini merupakan contoh kesalahan berdasarkan prinsip *proximity*.

- *Similarity*

Objek yang kelihatannya hampir sama tampak dalam satu grup.

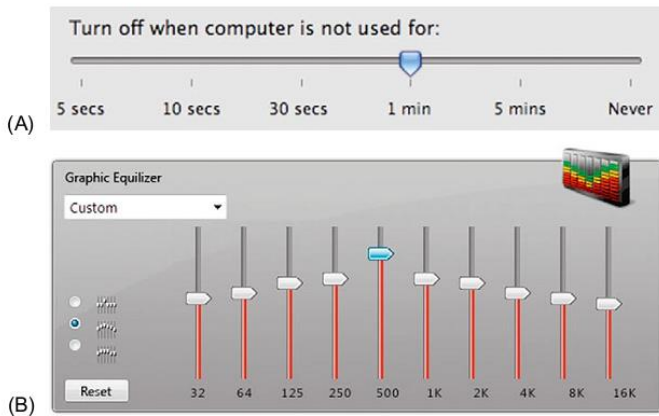


Gambar 2.6 Mac OS Page Setup Dialog Box

Pada Gambar 2.6 terlihat prinsip *Similarity* dan *Proximity* digunakan untuk mengelompokkan *orientation settings*.

- *Continuity*

Sistem visual kita mempunyai kecenderungan untuk mengartikan informasi ambigu salah satunya dengan menggunakan *continuity*.



Gambar 2.7 Mac OS Dan Component One

Pada Gambar 2.7 terlihat sebuah *slider* dengan *slot* tunggal dengan *handle* pada suatu tempat didalamnya.

- *Closure*

Berhubungan dengan *continuity*, sistem visual pada mata kita secara otomatis mencoba untuk menutup objek terbuka sehingga muncul persepsi sebagai sebuah objek daripada sebagai objek yang terpisah. Sering digunakan pada *icon* aplikasi.

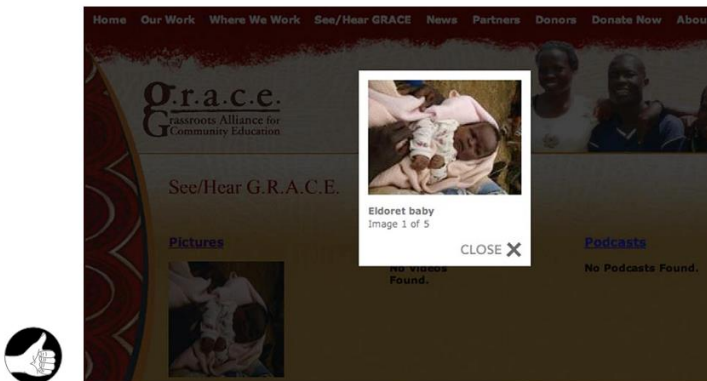


Gambar 2.8 Icon Dokumen

Pada Gambar 2.8 terlihat *icon* yang menunjukkan kumpulan dokumen.

- *Figure/Ground*

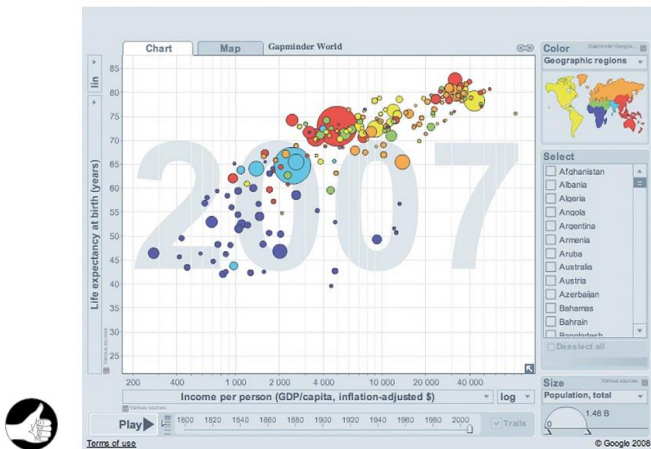
Prinsip ini mengatakan bahwa otak kita memisahkan tampilan visual ke dalam *figure* dan *ground*. Objek adalah sesuatu yang biasanya menjadi perhatian utama. Ketika dua buah objek *overlapping*, kita cenderung mengartikan objek yang lebih kecil sebagai *foreground* dan objek yang lebih besar sebagai *background*. Namun demikian untuk objek yang kompleks, *foreground* dan *background* lebih tergantung pada fokus manusia. Pada *user interface* atau desain *web*, prinsip *Figure/Ground* sering menggunakan *background* yang mengesankan untuk menyampaikan informasi mengenai: lokasi, tema, *brand* atau *mood* dari *content*. Selain itu, prinsip ini juga digunakan untuk menampilkan *popup* gambar. Lebih baik menampilkan informasi yang baru dalam bentuk *popup* dan informasi yang lama dalam bentuk *background* ketika dua buah informasi saling berhubungan dari pada menampilkan informasi yang baru dalam halaman yang terpisah dengan informasi yang lama.



Gambar 2.9 Pop Up Pada Graceusa.org

- *Common Fate*

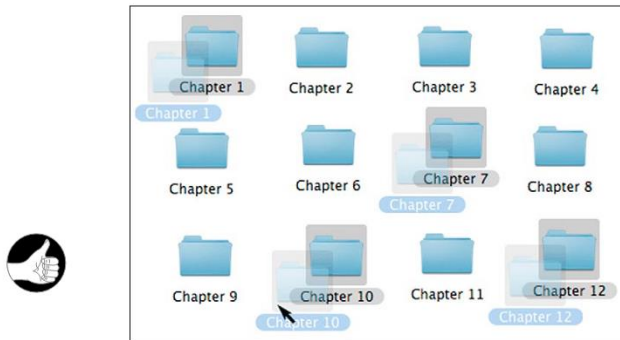
Prinsip-prinsip *gestalt* yang sebelumnya fokus untuk objek yang tidak bergerak, prinsip *Common fate* untuk objek yang bergerak. Prinsip ini mengatakan bahwa: objek yang bergerak bersama-sama dipersepsikan sebagai satu grup objek atau saling berhubungan.



Gambar 2.10 Gap-Minder Menganimasikan Titik-Titik Untuk Menunjukkan Negara-Negara Yang Mengalami Perkembangan Yang Mirip

- *Combined*

Pada dunia nyata, *Gestalt Principles* digunakan dengan kombinasi dari beberapa prinsip. *Similarity* dan *Common Fate*: ketika pengguna melakukan *drag folder* yang sudah dipilih maka *folder-folder* tersebut sama-sama di *highlight* dan mempunyai gerakan yang sama.



Gambar 2.11 Drag Folder Bersama

2.4.2 Heuristic Evaluation

Heuristic Usability atau yang juga dikenal sebagai *Heuristic Evaluation* adalah sistem evaluasi untuk *software* komputer berbasis pengguna. Sistem ini melibatkan *evaluator* untuk memberikan masukan yang kemudian dikategorikan dalam prinsip-prinsip heuristik. Meskipun dianggap sebagai metode informal dalam mengkaji kegunaan sebuah *software* atau aplikasi. Pendekatan yang diciptakan Nielsen di tahun 1990, adalah cara mengevaluasi yang cukup terpercaya dalam dunia New Media saat ini. Evaluasi Heuristik adalah panduan, prinsip umum, atau aturan yang dapat menuntun keputusan rancangan atau digunakan untuk mengkritik suatu keputusan yang sudah diambil[9].

Evaluasi Heuristik diusulkan oleh Nielsen dan Molich, hampir sama dengan *Cognitive Walkthrough* tetapi sedikit terstruktur dan sedikit terarah. Pada pendekatan ini, sekumpulan kriteria *usability* atau *heuristic* diidentifikasi dan perancangan dilaksanakan misalnya dimana kriteria dilanggar. Tujuan dari evaluasi heuristik adalah untuk memperbaiki perancangan secara efektif. *Evaluator* melakukan evaluasi melalui kinerja dari serangkaian tugas dengan perancangan dan dilihat kesesuaiannya dengan kriteria setiap tingkat. Jika ada kesalahan terdeteksi maka perancangan dapat ditinjau ulang untuk memperbaiki masalah ini

sebelum tingkat implementasi. Evaluasi Heuristik sangat baik digunakan sebagai teknik evaluasi desain, karena lebih mudah untuk menemukan atau menentukan masalah *usability* yang muncul.

Usability evaluation adalah proses yang melibatkan pengguna sehingga dapat mempelajari dan menggunakan produk guna tercapainya aspek-aspek kenyamanan pengguna seperti efektivitas, efisiensi, dan kepuasan pengguna terhadap sistem secara keseluruhan[10].

BAB III

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang perancangan dan pembuatan modul pengimplementasian Gestalt Principles pada aplikasi ClearRoute. Tugas Akhir ini diawali dari pembuatan desain sistem aplikasi, penggunaan prinsip Gestalt pada desain sistem, pembuatan desain user interface, pembuatan detail layout dan penggabungan alur antar layout. Setiap desain layout dirancang mengikuti prinsip Gestalt yang ditentukan. Untuk pengujian tahap alpha atau melalui developer dilakukan setidaknya dua kali untuk memastikan Gestalt Principles diimplementasikan dengan baik.

3.1 Deskripsi Umum

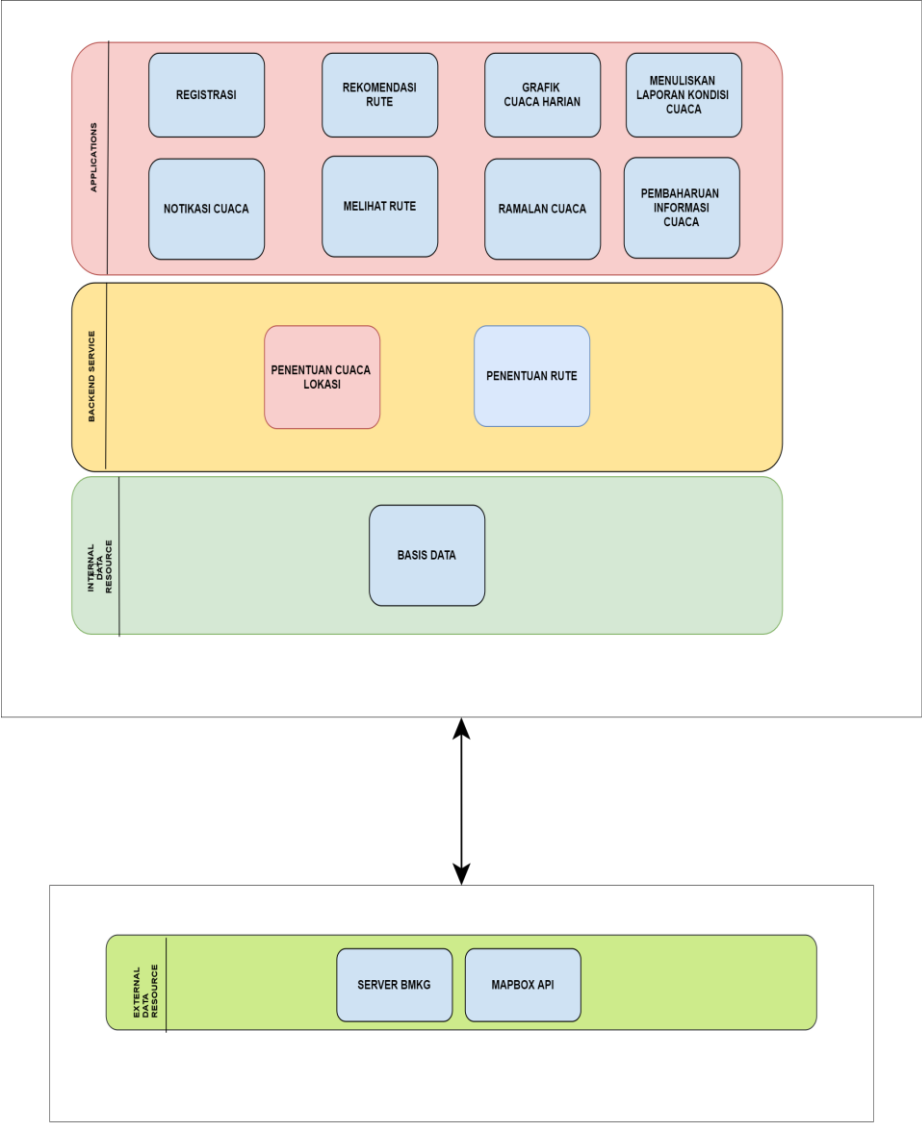
Tugas akhir ini merupakan sub-bagian dari aplikasi berbasis perangkat bergerak ClearRoute yang secara khusus menangani desain user interface dan user experience dengan didasarkan pada Gestalt Principles.

3.2 Arsitektur Sistem

Pada Gambar 3.1 ditunjukkan rancangan arsitektur aplikasi Clearroute secara umum. Dimana sistem yang akan dibangun terdiri dari 3 bagian yaitu:

- a. Applications*
- b. Backend-Services*
- c. Internal data resources*
- d. External data resources*

Keempat bagian tersebut saling berhubungan satu sama lainnya agar aplikasi Clearroute yang dibangun dapat bekerja dengan baik. Pada *external data resources* terdapat sumber data yang berasal dari BMKG sehingga sistem yang dibangun tidak menangani pengolahan data BMKG dan hanya menerima informasi cuaca yang sudah jadi dari pusat data BMKG.



Gambar 3.1 Diagram Arsitektur Aplikasi ClearRoute

3.2.1 Applications

Bagian ini merupakan fitur-fitur yang terdapat pada aplikasi Clearroute itu sendiri. Pada aplikasi sendiri terdapat beberapa fitur yang dimiliki antara lain:

- *Registrasi / Sign Up*
- *Login / Sign In*
- Melihat profil
- Mengedit profil
- Rekomendasi rute
- Grafik cuaca harian
- Menuliskan laporan cuaca
- Ramalan cuaca terkini
- Pembaharuan informasi cuaca

Pada aplikasi ini diterapkan dua pemanfaatan lokasi geografis pengguna, yaitu :

1) *Location Based Service*

Data lokasi pengguna sangatlah vital untuk dieksplotasi lebih lanjut. Dari data lokasi ini dapat digunakan untuk memudahkan kegiatan sehari-hari pengguna terutama pengguna kendaraan bermotor baik roda dua maupun roda empat. Data lokasi pengguna dimanfaatkan aplikasi ke dalam tiga fitur utamanya, yaitu menampilkan ramalan cuaca pada hari itu di lokasi pengguna berada, menampilkan grafik ramalan setiap hari nya dengan rentang waktu tiga hari sebelum dan sesudahnya, dan yang utama dapat melakukan navigasi pengguna agar sampai ke tujuan dengan resiko terkena hujan seminimal mungkin.

Pada fitur pertama, aplikasi akan mengirimkan data posisi *latitude* dan *longitude* dirinya untuk dikirim ke *backend service*. Kemudian dari *backend service* akan direspon dengan mengirimkan data ramalan selama satu hari penuh.

Pada fitur kedua, aplikasi akan mengirimkan data lokasi sama halnya pada fitur pertama dan akan direspon dengan mengirimkan data cuaca tiga hari sebelum dan sesudahnya.

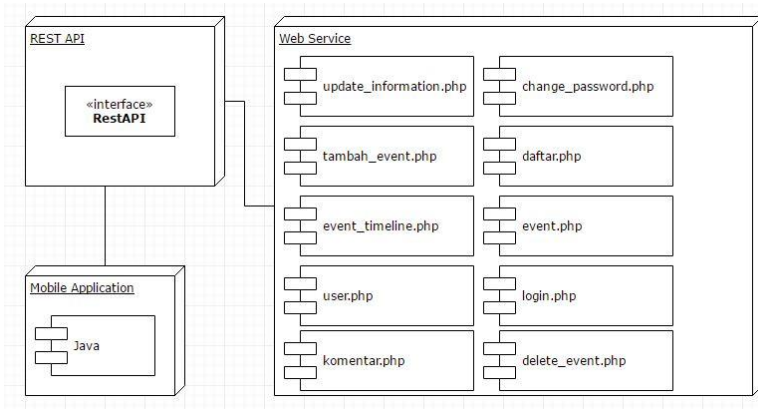
Pada fitur ketiga, aplikasi akan mengirimkan data *latitude* dan *longitude* lokasi dirinya dan lokasi tujuannya. Kemudian aplikasi akan mendapatkan respon berupa data rute untuk ditampilkan di peta.

2) *Community Based Service*

Untuk melengkapi fitur-fitur dari *Location Based Service*, tetap dibutuhkan data dan informasi dari manusia. Karena pada dasarnya manusia adalah makhluk sosial yang akan saling memberi informasi satu sama lain. Meskipun data dari BMKG adalah data resmi untuk digunakan di Indonesia, akurasi dan representasi datanya masih kurang dimanfaatkan oleh masyarakat Indonesia. Pada aplikasi ini, *community based service*-nya lebih mirip dengan Instagram yang dimodifikasi sesuai peruntukannya. Layaknya sosial media, di dalam aplikasi ini pengguna dapat mengunggah laporan berita disekitarnya dan pengguna lain dapat melihatnya. Selain itu, semua postingan pengguna akan tersimpan dalam akun mereka dan tentunya bisa dilihat oleh pengguna lain.

3.2.2 *Backend Services*

Bagian ini merupakan sistem yang akan menangani dua hal utama pada *server* aplikasi Clearroute, yaitu pembangunan autentifikasi aplikasi, dan proses sosial pada sistem *community based service*.



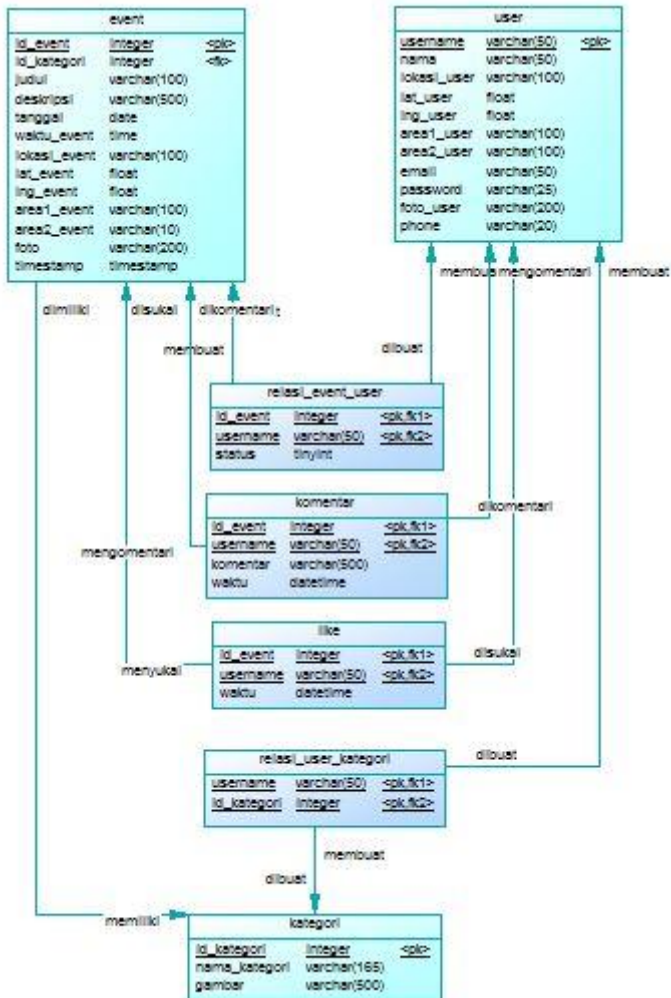
Gambar 3.2 Diagram Komponen *Backend-Services*

Pada Gambar 3.2 terlihat bagian ini juga dilakukan proses perputaran data untuk *community based service* layaknya pembuatan akun hingga melihat berita cuaca pengguna lain. Aplikasi perangkat bergerak ini menggunakan RestAPI dan Retrofit untuk mengakses *file web service* di *server*.

3.2.3 *Internal Data Resources*

Bagian ini bertanggung jawab untuk mengatur basis data yang akan digunakan oleh aplikasi ClearRoute. Jenis basis data yang digunakan adalah MySQL sehingga basis data tersebut dapat mengolah data yang akan digunakan pada tugas akhir ini.

Gambar 3.3 memperlihatkan seluruh isi basis data yang digunakan untuk menyelesaikan tujuan dari Tugas Akhir ini.



Gambar 3.3 Desain PDM Basis Data

3.2.4 External Data Resource

Tabel 3.1 Data Kondisi Cuaca yang Sedang Terjadi

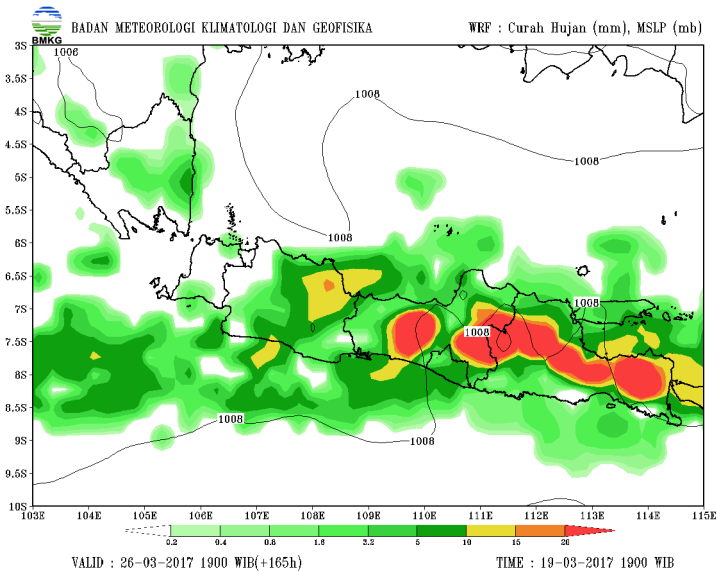
lokasi	lat	lon	curan	kategori
201701091250	226	1	0.006	1
201701091250	225	1	0.006	1
201701091250	224	1	0.006	1
201701091250	223	1	0.006	1
201701091250	222	1	0.004666667	0
201701091250	221	1	0.005333333	0
201701091250	220	1	0.006	1
201701091250	219	1	0.004	0
201701091250	218	1	0.092	2
201701091250	217	1	0.006	1
201701091250	216	1	0.085777778	2
201701091250	215	1	0.006	1
201701091250	214	1	0.003333333	0
201701091250	213	1	0.042222222	2

Bagian ini merupakan sumber data luar yang digunakan oleh aplikasi Clearroute. Pada pengerjaan tugas akhir ini sumber data yang digunakan ialah data cuaca yang dimiliki oleh BMKG. Terdapat dua jenis data yang akan digunakan yaitu data kondisi cuaca yang sedang terjadi dan data ramalan cuaca yang akan terjadi seperti yang ditunjukkan oleh Tabel 3.1. Pada pengumpulan data kondisi cuaca yang sedang terjadi, BMKG menggunakan alat-alat yang terdapat pada pos hujan di mana pos hujan tersebut ditempatkan di seluruh kelurahan pada Kota Surabaya. Data-data tersebut kemudian diolah oleh pihak BMKG dan informasi mengenai kondisi cuaca yang sedang terjadi akan dikirimkan melalui protokol HTTP *Request* menggunakan format data XML.

Untuk data ramalan cuaca BMKG mengeluarkan informasinya setiap pukul 00.00 WIB dan data ramalan BMKG tersebut berlaku setiap 3 jam sehingga untuk data ramalan selama 1 hari BMKG akan mengeluarkan 8 data. Data yang dikeluarkan BMKG untuk menginformasikan ramalan cuaca adalah data citra

sehingga diperlukan suatu proses untuk mengolah data citra tersebut untuk mendapatkan informasi yang diinginkan.

Gambar 3.4 menunjukkan kondisi ramalan cuaca yang akan terjadi pada tanggal 26 Maret 2017. Pada Gambar 3.5, Tabel 3.2, dan Tabel 3.3 ditunjukkan keterangan kondisi cuaca yang digambarkan pada peta ramalan cuaca yang dikeluarkan oleh BMKG.



Gambar 3.4 Data Citra Ramalan Cuaca BMKG



Gambar 3.5 Keterangan Kondisi Cuaca

Tabel 3.2 Skala Keterangan Kondisi Cuaca

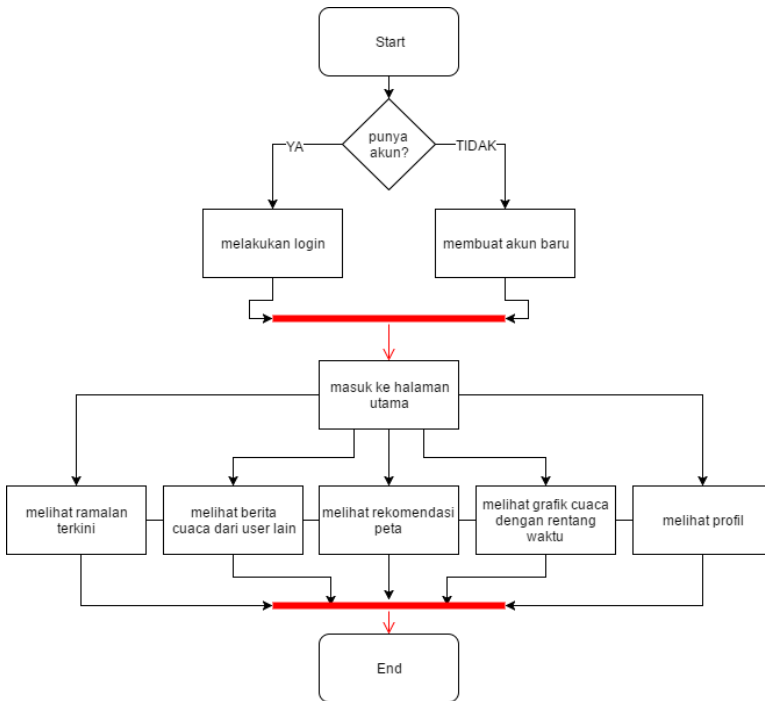
Skala	Kondisi cuaca
0.2-5	Berawan
5-10	Hujan ringan
10-15	Hujan sedang
15-20	Hujan deras
20 >	Hujan badai

Tabel 3.3 Kode RGB Keterangan Kondisi Cuaca

Kode RGB	Kondisi Cuaca
#b4faaa	Berawan
#96f58c	Berawan
#78f573	Berawan
#50f050	Berawan
#37d23c	Berawan
#0fa00f	Hujan Ringan
#fa0f	Hujan Ringan
#e6dc32	Hujan Sedang
#f08228	Hujan Deras
#fa3c3c	Hujan Badai

3.3 Analisis Proses Bisnis

Proses bisnis yang digunakan dalam aplikasi ini adalah proses bisnis yang umum terjadi pada aplikasi perangkat bergerak kategori sosial dan navigasi pengguna. Dengan mengacu pada tinjauan pustaka, alur proses bisnis yang didapatkan dapat dilihat pada diagram alir proses berikut.



Gambar 3.6 Diagram Alir Proses Bisnis

Pada Gambar 3.6, setiap pengguna diminta untuk memiliki akun terlebih dahulu sebelum dapat menikmati berbagai fitur menarik pada aplikasi ini. Setelah memiliki akun, pengguna dapat masuk ke halaman utama yang sudah dalam bentuk *Tabbed Page* berisi lima fitur utama aplikasi ini.

3.4 Analisis Kebutuhan Sistem

Analisis spesifikasi kebutuhan dalam aplikasi ini mencakup kebutuhan fungsional. Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Kebutuhan fungsional didapatkan dari analisis proses bisnis yang

telah dijelaskan sebelumnya. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.4.

Tabel 3.4 Daftar Kebutuhan Fungsional Sistem

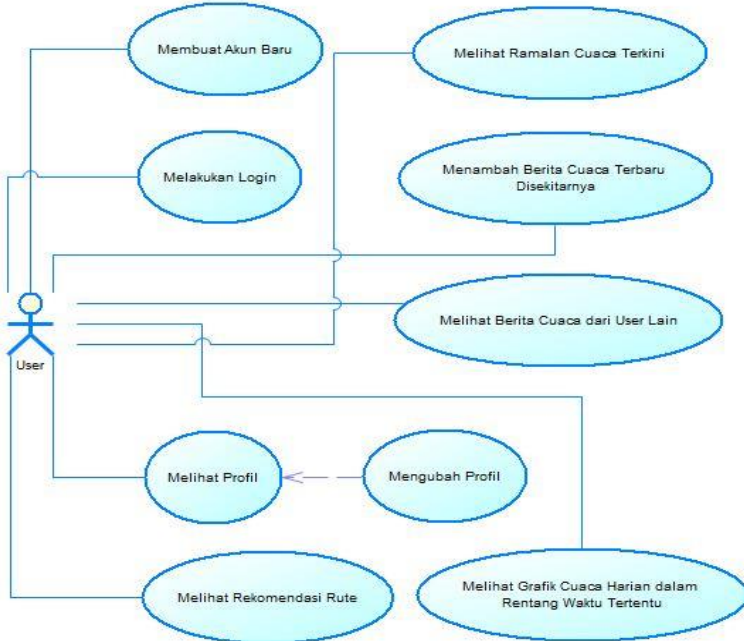
Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	Mendaftar Akun	Melakukan pendaftaran untuk dapat menggunakan fitur aplikasi lainnya
F-002	Melakukan <i>Login</i>	Memasukkan data akun untuk memasuki halaman utama aplikasi
F-003	Melihat Profil	Melihat data-data pada profil pengguna
F-004	Mengubah Profil	Mengubah data-data pada profil pengguna
F-005	Melihat Ramalan Cuaca	Melihat data ramalan cuaca untuk satu hari penuh
F-006	Melihat Berita Cuaca	Melihat berita cuaca yang dibuat oleh pengguna lainnya
F-007	Membuat Berita Cuaca	Membuat berita cuaca terbaru di sekitar pengguna
F-008	Melihat Rekomendasi Rute	Melihat rekomendasi rute dengan minim terkena hujan
F-009	Melihat Grafik	Melihat grafik cuaca dengan rentang waktu tertentu

3.5 Analisis Aktor

Aktor mendefinisikan *stakeholder* yang terlibat dan berinteraksi langsung dengan sistem. *Stakeholder* ini bisa berupa

manusia maupun sistem atau perangkat lunak yang lain. *Stake holder* yang terdapat pada sistem ini hanya sebagai pengguna.

Pengguna tersebut merupakan mayoritas ditaksir seorang pengendara motor roda dua maupun roda empat.



Gambar 3.7 Diagram Kasus Penggunaan

3.6 Perancangan Sistem

Tahap perancangan sistem dibagi menjadi beberapa bagian yaitu kasus penggunaan dan perancangan antarmuka aplikasi.

3.6.1 Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan sebelumnya pada spesifikasi kebutuhan fungsional dan identifikasi aktor. Diagram kasus

penggunaan dapat dilihat pada Gambar 3.7 dan kode kasus penggunaan pada Tabel 3.5.

Tabel 3.5 Keterangan Kode Kasus Penggunaan

Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	Mendaftar Akun
UC-002	Melakukan <i>Login</i>
UC-003	Melihat Profil
UC-004	Mengubah Profil
UC-005	Melihat Ramalan Terbaru
UC-006	Melihat Berita Cuaca
UC-007	Membuat Berita Cuaca
UC-008	Melihat Rekomendasi Rute
UC-009	Melihat Grafik

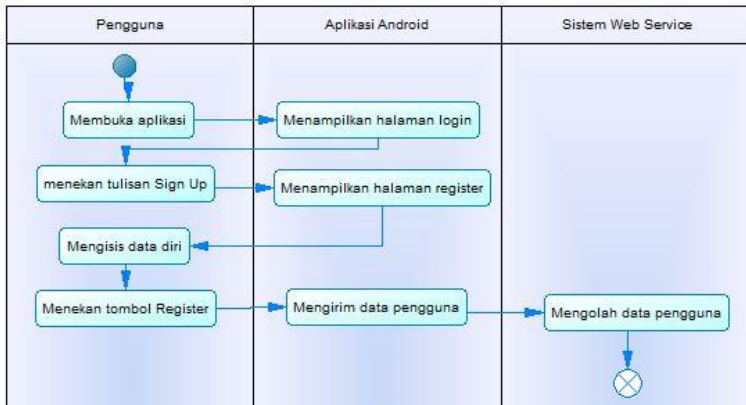
3.6.1.1 Kasus Penggunaan Mendaftar Akun

Pada kasus penggunaan ini, pengguna mendaftar akun untuk bisa masuk ke dalam halaman utama aplikasi. Spesifikasi kasus penggunaan mendaftar akun aplikasi dapat dilihat pada Tabel 3.6 dan diagram aktivitas pada Gambar 3.8.

Tabel 3.6 Spesifikasi Kasus Penggunaan Mendaftar Akun

Kode	UC-001
Nama	Mendaftar Akun
Aktor	Pengguna
Deskripsi	Pengguna mendaftar akun pada aplikasi ClearRoute
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada <i>layout</i> halaman <i>registrasi</i>

Kondisi Akhir	Pengguna berhasil mendapat akun untuk memasuki halaman utama
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Aplikasi menampilkan halaman <i>Login</i> 3. Pengguna menekan tulisan <i>Sign Up</i> 4. Aplikasi menampilkan halaman <i>Register</i> 5. Pengguna mengisi data diri sesuai ketentuan 6. Pengguna menekan tombol <i>Register</i> 7. Aplikasi mengirim data pengguna ke sistem 8. Sistem mengolah data pengguna 	
Alur Kejadian Alternatif	
Data diri yang diisi pengguna tidak sesuai ketentuan	
<ol style="list-style-type: none"> 1. Aplikasi menunjukkan <i>warning</i> berupa tulisan merah di kolom yang tidak sesuai 2. Pengguna melanjutkan pengisian data 	
Data email sudah digunakan atau terdaftar	
<ol style="list-style-type: none"> 1. Aplikasi menunjukkan <i>warning</i> berupa tulisan merah di kolom email 2. Pengguna melanjutkan pengisian data atau kembali ke halaman <i>login</i> 	



Gambar 3.8 Diagram Aktivitas Mendaftar Akun

3.6.1.2 Kasus Penggunaan Melakukan *Login*

Pada kasus penggunaan ini, pengguna melakukan *login* untuk bisa masuk ke dalam halaman utama aplikasi. Spesifikasi kasus penggunaan melakukan *login* dapat dilihat pada Tabel 3.7 dan diagram aktivitas pada Gambar 3.9.

Tabel 3.7 Spesifikasi Kasus Penggunaan Melakukan *Login*

Kode	UC-002
Nama	Melakukan <i>Login</i>
Aktor	Pengguna
Deskripsi	Pengguna melakukan <i>login</i> pada aplikasi ClearRoute
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada <i>layout</i> halaman <i>login</i>
Kondisi Akhir	Pengguna berhasil memasuki halaman utama
Alur Kejadian Normal	
1. Pengguna membuka aplikasi	
2. Aplikasi menampilkan halaman <i>Login</i>	

3. Pengguna mengisi data diri sesuai ketentuan
4. Pengguna menekan tombol *Login*
5. Aplikasi mengirim data pengguna ke sistem
6. Sistem mengolah data pengguna

Alur Kejadian Alternatif

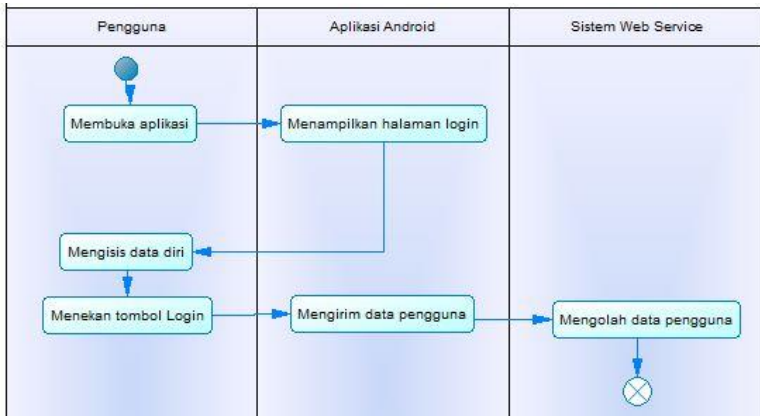
Data yang diisi pengguna tidak sesuai ketentuan

1. Aplikasi menunjukkan *warning* berupa tulisan merah di kolom yang tidak sesuai

2. Pengguna melanjutkan pengisian data

Pengguna lupa *password* dari *email* yang terdaftar

1. Pengguna menekan tombol *Forgot Password*
2. Aplikasi menampilkan halaman *Forgot Password*
3. Pengguna memasukkan data nomor *handphone*
4. Aplikasi mengirim data pengguna ke sistem
5. Sistem mengolah data dan mengirimkan kode pin ke nomor *handphone* yang terdaftar
6. Pengguna memasukkan data kode pin
7. Aplikasi mengirim data kode pin ke sistem
8. Sistem mengolah data
9. Jika kode pin tidak sesuai, maka aplikasi akan memberikan *warning* berupa *snackbar* dan akan menampilkan halaman *login*
10. Jika kode pin sesuai, maka aplikasi akan menampilkan halaman *reset password*
11. Pengguna melakukan pengisian data *password* baru
12. Aplikasi mengirim data *password* ke sistem
13. Sistem mengolah data pengguna
14. Aplikasi menampilkan halaman *login* dengan notifikasi berhasil
15. Pengguna dapat melanjutkan pengisian data



Gambar 3.9 Diagram Aktivitas Melakukan Login

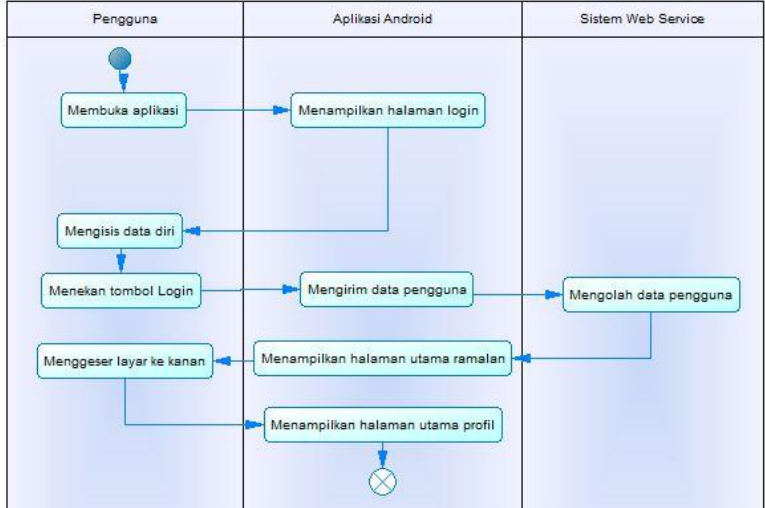
3.6.1.3 Kasus Penggunaan Melihat Profil

Pada kasus penggunaan ini, pengguna melihat profil dirinya pada halaman utama aplikasi. Spesifikasi kasus penggunaan melihat profil dapat dilihat pada Tabel 3.8 dan diagram aktivitas pada Gambar 3.10.

Tabel 3.8 Spesifikasi Kasus Penggunaan Melihat Profil

Kode	UC-003
Nama	Melihat Profil
Aktor	Pengguna
Deskripsi	Pengguna melihat profil dirinya pada aplikasi ClearRoute
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada <i>layout</i> halaman login
Kondisi Akhir	Pengguna berada pada <i>layout</i> halaman profil
Alur Kejadian Normal	
1. Pengguna membuka aplikasi 2. Aplikasi menampilkan halaman <i>Login</i> 3. Pengguna mengisi data diri sesuai ketentuan	

4. Pengguna menekan tombol <i>Login</i>
5. Aplikasi mengirim data pengguna ke sistem
6. Sistem mengolah data pengguna
7. Aplikasi menampilkan halaman utama ramalan
8. Pengguna menggeser layar ke kanan empat kali
9. Aplikasi menampilkan halaman utama profil

| Alur Kejadian Alternatif |
| Halaman aplikasi tidak dapat digeser |
| 1. Pengguna menekan *icon* profil pada layar ponsel |


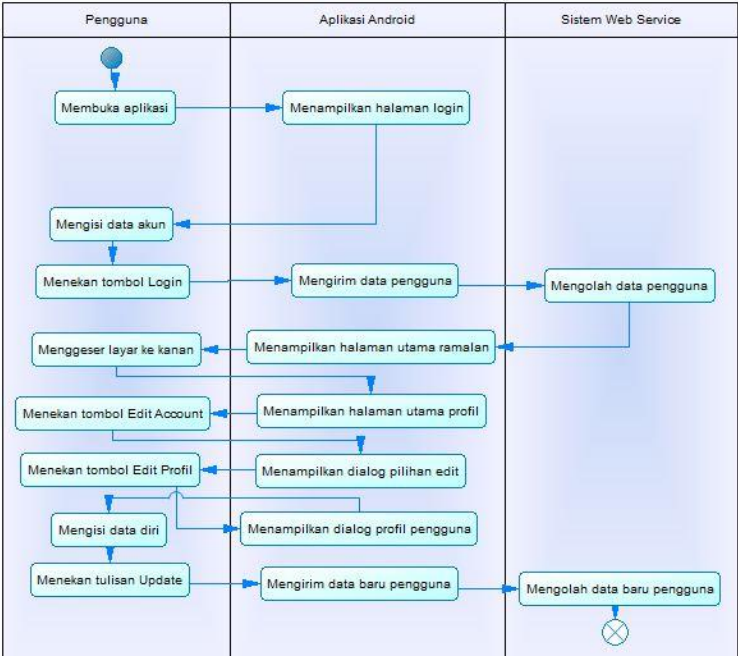
Gambar 3.10 Diagram Aktivitas Melihat Profil

3.6.1.4 Kasus Penggunaan Mengubah Profil

Pada kasus penggunaan ini, pengguna mengubah profil dirinya pada halaman utama aplikasi. Spesifikasi kasus penggunaan mengubah profil dapat dilihat pada Tabel 3.9 dan diagram aktivitas pada Gambar 3.11.

Tabel 3.9 Spesifikasi Kasus Penggunaan Mengubah Profil

Kode	UC-004
Nama	Mengubah Profil
Aktor	Pengguna
Deskripsi	Pengguna mengubah profil dirinya pada aplikasi ClearRoute
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada <i>layout</i> halaman login
Kondisi Akhir	Pengguna berhasil mengubah data profilnya
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Aplikasi menampilkan halaman <i>Login</i> 3. Pengguna mengisi data diri sesuai ketentuan 4. Pengguna menekan tombol <i>Login</i> 5. Aplikasi mengirim data pengguna ke sistem 6. Sistem mengolah data pengguna 7. Aplikasi menampilkan halaman utama ramalan 8. Pengguna menggeser layar ke kanan empat kali 9. Aplikasi menampilkan halaman utama profil 10. Pengguna menekan tombol <i>Edit Account</i> 11. Aplikasi menampilkan dialog pilihan edit 12. Pengguna menekan tombol Edit Profil 13. Aplikasi menampilkan dialog profil pengguna 14. Pengguna mengisi data diri yang baru 15. Pengguna menekan tulisan <i>Update</i> 16. Aplikasi mengirim data pengguna ke sistem 17. Sistem mengolah data pengguna 	
Alur Kejadian Alternatif	
<p>Data diri pengguna tidak sesuai dengan ketentuan</p> <ol style="list-style-type: none"> 1. Aplikasi menampilkan <i>warning</i> berupa tulisan merah di kolom yang tidak sesuai 2. Pengguna melanjutkan pengisian data 	



Gambar 3.11 Diagram Aktivitas Mengubah Profil

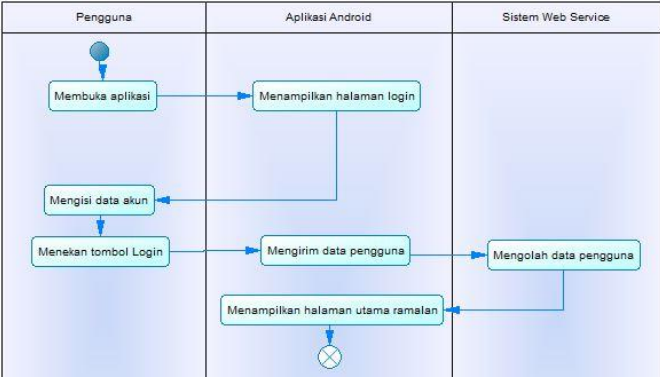
3.6.1.5 Kasus Penggunaan Melihat Ramalan Terkini

Pada kasus penggunaan ini, pengguna melihat ramalan terkini pada halaman utama aplikasi. Spesifikasi kasus penggunaan melihat ramalan terkini dapat dilihat pada Tabel 3.10 dan diagram aktivitas pada Gambar 3.12.

Tabel 3.10 Spesifikasi Kasus Penggunaan Melihat Ramalan Terkini

Kode	UC-005
Nama	Melihat Ramalan Terkini
Aktor	Pengguna

Deskripsi	Pengguna melihat ramalan terkini pada aplikasi ClearRoute
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada <i>layout</i> halaman login
Kondisi Akhir	Pengguna berada pada <i>layout</i> halaman ramalan
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Aplikasi menampilkan halaman <i>Login</i> 3. Pengguna mengisi data diri sesuai ketentuan 4. Pengguna menekan tombol <i>Login</i> 5. Aplikasi mengirim data pengguna ke sistem 6. Sistem mengolah data pengguna 7. Aplikasi menampilkan halaman utama ramalan 	
Alur Kejadian Alternatif	
<p>Ponsel tidak mengaktifkan GPS</p> <ol style="list-style-type: none"> 1. Aplikasi menampilkan notifikasi untuk mengaktifkan GPS 2. Pengguna setuju untuk mengaktifkan GPS 3. Aplikasi menampilkan halaman setelan pada ponsel untuk pengaktifan GPS 4. Pengguna mengaktifkan GPS 5. Aplikasi kembali pada halaman ramalan 	



Gambar 3.12 Diagram Aktivitas Melihat Ramalan Terkini

3.6.1.6 Kasus Penggunaan Melihat Berita Cuaca

Pada kasus penggunaan ini, pengguna melihat berita cuaca pada halaman utama aplikasi. Spesifikasi kasus penggunaan melihat berita cuaca dapat dilihat pada Tabel 3.11 dan diagram aktivitas pada Gambar 3.13.

Tabel 3.11 Spesifikasi Kasus Penggunaan Melihat Berita Cuaca

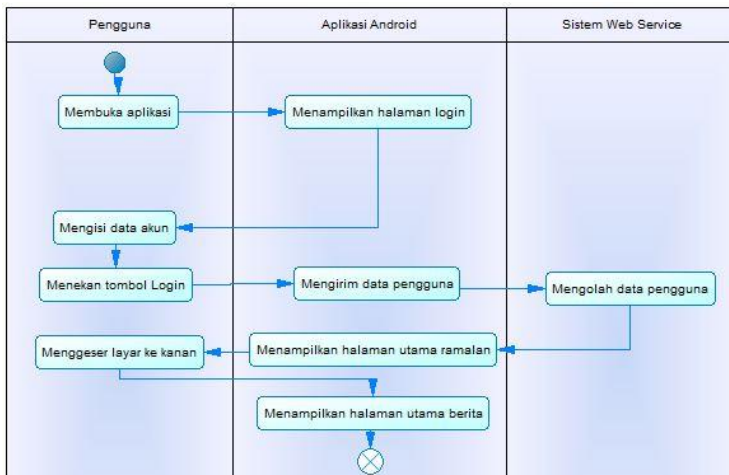
Kode	UC-006
Nama	Melihat Berita Cuaca
Aktor	Pengguna
Deskripsi	Pengguna melihat berita cuaca pada aplikasi ClearRoute
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada <i>layout</i> halaman login
Kondisi Akhir	Pengguna berada pada <i>layout</i> halaman berita
Alur Kejadian Normal	
1. Pengguna membuka aplikasi	
2. Aplikasi menampilkan halaman <i>Login</i>	
3. Pengguna mengisi data diri sesuai ketentuan	

4. Pengguna menekan tombol *Login*
5. Aplikasi mengirim data pengguna ke sistem
6. Sistem mengolah data pengguna
7. Aplikasi menampilkan halaman utama ramalan
8. Pengguna menggeser layar ke kanan satu kali
9. Aplikasi menampilkan halaman utama berita

Alur Kejadian Alternatif

Halaman aplikasi tidak dapat digeser

1. Pengguna menekan *icon* berita pada layar ponsel



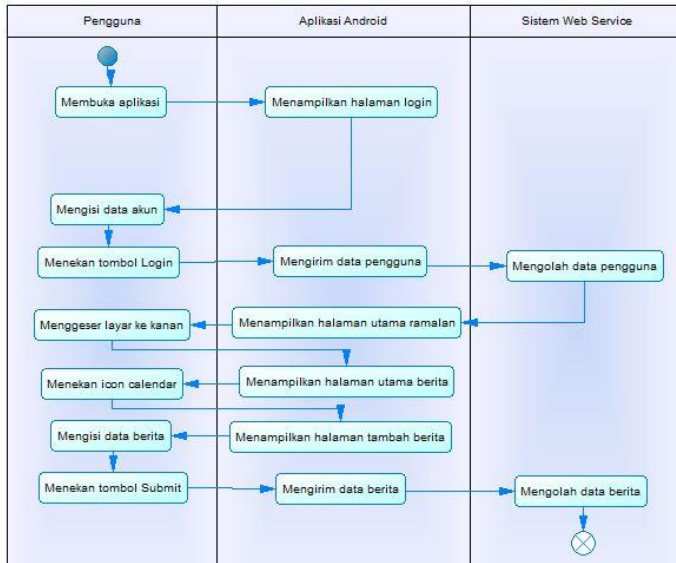
Gambar 3.13 Diagram Aktivitas Melihat Berita Cuaca

3.6.1.7 Kasus Penggunaan Membuat Berita Cuaca

Pada kasus penggunaan ini, pengguna membuat berita cuaca pada halaman utama aplikasi. Spesifikasi kasus penggunaan membuat berita cuaca dapat dilihat pada Tabel 3.12 dan diagram aktivitas pada Gambar 3.14.

Tabel 3.12 Spesifikasi Kasus Penggunaan Membuat Berita Cuaca

Kode	UC-006
Nama	Membuat Berita Cuaca
Aktor	Pengguna
Deskripsi	Pengguna membuat berita cuaca pada aplikasi ClearRoute
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada <i>layout</i> halaman login
Kondisi Akhir	Pengguna berhasil membuat berita cuaca
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Aplikasi menampilkan halaman <i>Login</i> 3. Pengguna mengisi data diri sesuai ketentuan 4. Pengguna menekan tombol <i>Login</i> 5. Aplikasi mengirim data pengguna ke sistem 6. Sistem mengolah data pengguna 7. Aplikasi menampilkan halaman utama ramalan 8. Pengguna menggeser layar ke kanan satu kali 9. Aplikasi menampilkan halaman utama berita 10. Pengguna menekan <i>icon calendar</i> pada <i>toolbar</i> 11. Aplikasi menampilkan halaman tambah berita 12. Pengguna mengisi data berita sesuai ketentuan 13. Pengguna menekan tombol <i>Submit</i> 14. Aplikasi mengirim data berita ke sistem 15. Sistem mengolah data berita 	
Alur Kejadian Alternatif	
Tidak ada.	



Gambar 3.14 Diagram Aktivitas Membuat Berita Cuaca

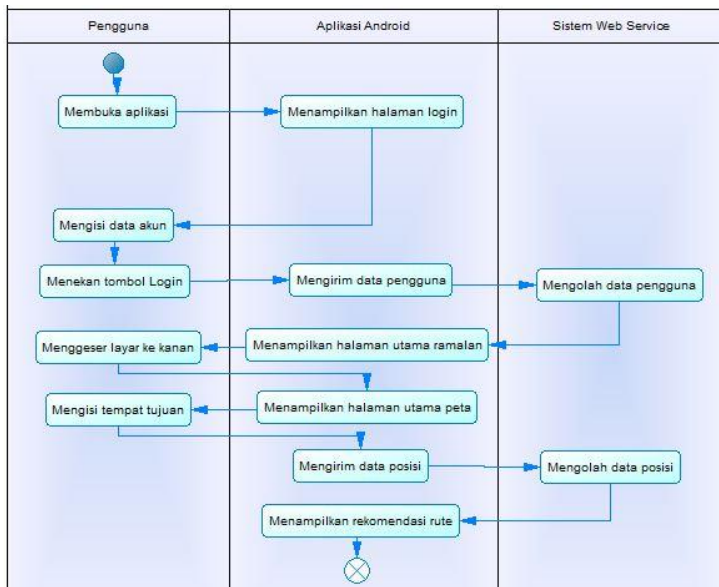
3.6.1.8 Kasus Penggunaan Melihat Rekomendasi Rute

Pada kasus penggunaan ini, pengguna melihat rekomendasi rute pada halaman utama aplikasi. Spesifikasi kasus penggunaan melihat rekomendasi rute dapat dilihat pada Tabel 3.13 dan diagram aktivitas pada Gambar 3.15.

Tabel 3.13 Spesifikasi Kasus Penggunaan Melihat Rekomendasi Rute

Kode	UC-008
Nama	Melihat Rekomendasi Rute
Aktor	Pengguna
Deskripsi	Pengguna melihat rekomendasi rute pada aplikasi ClearRoute
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada <i>layout</i> halaman login

Kondisi Akhir	Pengguna berada pada <i>layout</i> halaman peta
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Aplikasi menampilkan halaman <i>Login</i> 3. Pengguna mengisi data diri sesuai ketentuan 4. Pengguna menekan tombol <i>Login</i> 5. Aplikasi mengirim data pengguna ke sistem 6. Sistem mengolah data pengguna 7. Aplikasi menampilkan halaman utama ramalan 8. Pengguna menggeser layar ke kanan dua kali 9. Aplikasi menampilkan halaman utama peta 10. Pengguna mengisi tempat tujuan 11. Aplikasi mengirim data posisi 12. Sistem mengolah data posisi 13. Aplikasi menampilkan rekomendasi rute 	
Alur Kejadian Alternatif	
Halaman aplikasi tidak dapat digeser	
<ol style="list-style-type: none"> 1. Pengguna menekan <i>icon</i> peta pada layar ponsel 	
Data alamat tujuan tidak sesuai dengan ketentuan	
<ol style="list-style-type: none"> 1. Aplikasi menampilkan <i>warning</i> berupa tulisan merah di kolom yang tidak sesuai 2. Pengguna melanjutkan pengisian data 	



Gambar 3.15 Diagram Aktivitas Melihat Rekomendasi Rute

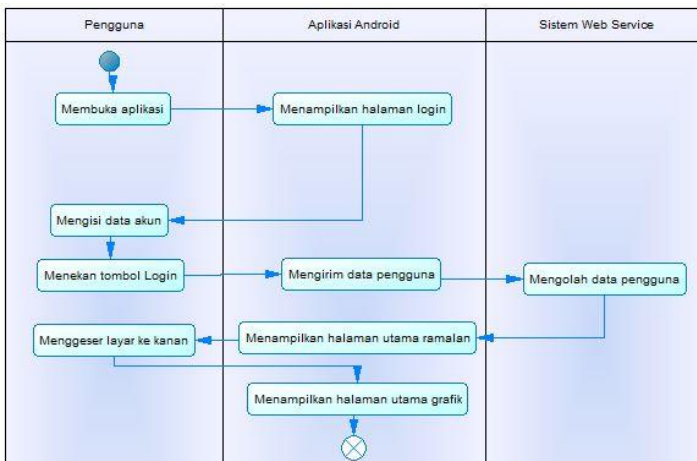
3.6.1.9 Kasus Penggunaan Melihat Grafik

Pada kasus penggunaan ini, pengguna melihat grafik pada halaman utama aplikasi. Spesifikasi kasus penggunaan melihat grafik dapat dilihat pada Tabel 3.14 dan diagram aktivitas pada Gambar 3.16.

Tabel 3.14 Spesifikasi Kasus Penggunaan Melihat Grafik

Kode	UC-009
Nama	Melihat Grafik
Aktor	Pengguna
Deskripsi	Pengguna melihat grafik pada aplikasi ClearRoute
Tipe	Fungsional

Kondisi Awal	Pengguna berada pada <i>layout</i> halaman login
Kondisi Akhir	Pengguna berada pada <i>layout</i> halaman grafik
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Aplikasi menampilkan halaman <i>Login</i> 3. Pengguna mengisi data diri sesuai ketentuan 4. Pengguna menekan tombol <i>Login</i> 5. Aplikasi mengirim data pengguna ke sistem 6. Sistem mengolah data pengguna 7. Aplikasi menampilkan halaman utama ramalan 8. Pengguna menggeser layar ke kanan satu kali 9. Aplikasi menampilkan halaman utama grafik 	
Alur Kejadian Alternatif	
Halaman aplikasi tidak dapat digeser	
<ol style="list-style-type: none"> 1. Pengguna menekan <i>icon</i> grafik pada layar ponsel 	



Gambar 3.16 Diagram Aktivitas Melihat Grafik

3.6.2 Antarmuka Aplikasi

Perancangan antarmuka merupakan hal yang penting dalam melakukan perancangan aplikasi. Antarmuka pengguna yang

berhubungan langsung dengan aktor haruslah memiliki kemudahan-kemudahan dan tampilan yang efisien bagi penggunaanya.

3.6.2.1 Antarmuka Halaman *Register*

Pada antarmuka tersebut terdapat 4 data pengguna yang perlu diisi, yaitu nama, *email*, nomor *handphone* dan *password* akun, dan tombol untuk mengirim data-data tadi ke sistem.



Gambar 3.17 Antarmuka Halaman *Register*

Di dalam halaman *register* direncanakan akan menggunakan prinsip *Proximity* untuk kelompok kolom isian oleh pengguna.

3.6.2.2 Antarmuka Halaman *Login*

Pada antarmuka tersebut terdapat 2 data pengguna yang perlu diisi, yaitu *email* dan *password*, dan tombol untuk mengirim data-data tadi ke sistem, serta beberapa link menuju halaman *register* maupun halaman pemulihan *password*.

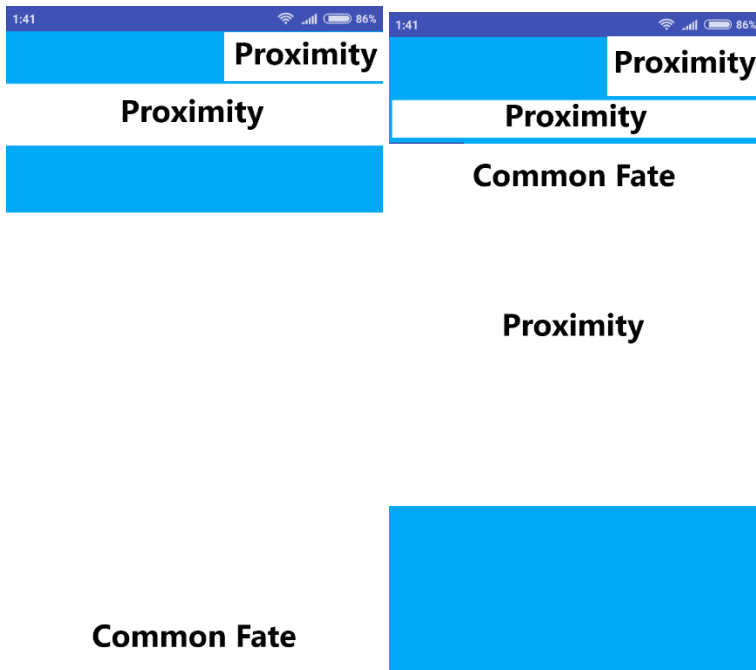


Gambar 3.18 Antarmuka Halaman *Login*

Di dalam halaman *login* akan ada dua pengelompokan, yaitu pengelompokan kolom isian dan pengelompokan *link* lanjutan. Pengelompokan-pengelompokan tersebut direncanakan akan menerapkan prinsip *Proximity* didalamnya.

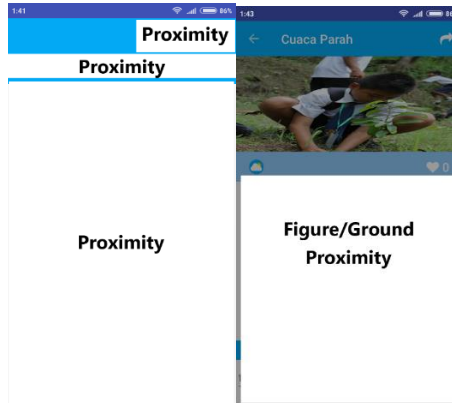
3.6.2.3 Antarmuka Halaman Utama

Pada antarmuka tersebut terdapat 5 halaman yang ditampilkan dalam satu halaman menggunakan cara *Tabbed Page*. Pada sub-halaman pertama terdapat tampilan kondisi cuaca untuk satu hari penuh yang terbagi pada beberapa waktu. Pada sub-halaman kedua terdapat tampilan daftar berita yang dapat di *scroll* kebawah. Pada sub-halaman ketiga terdapat tampilan peta dengan *marker* berupa *marker* orang dan *marker* cuaca. Pada sub-halaman keempat terdapat tampilan daftar grafik dengan *marker* cuaca. Pada sub-halaman kelima terdapat tampilan profil pengguna dengan daftar berita yang pernah dibuat.



Gambar 3.19 Antarmuka Halaman Cuaca

Di dalam halaman cuaca akan diterapkan dua prinsip *Gestalt*, yaitu *Proximity* dan *Common Fate*. *Common Fate* digunakan pada animasi penunjukan detail cuaca harian. Sedangkan *Proximity* digunakan pada pengelompokan cuaca harian, *icon toolbar*, dan *icon* halaman utama.



Gambar 3.20 Antarmuka Halaman Berita

Pada halaman berita, prinsip *Gestalt* yang digunakan ada dua, yaitu *Figure/Ground* dan *Proximity*. *Figure/Ground* digunakan pada saat pengguna akan membagikan berita. Sedangkan, *Proximity* digunakan pada pengelompokan daftar media berbagi, daftar berita, *icon toolbar*, dan *icon* halaman utama.



Figure/Ground

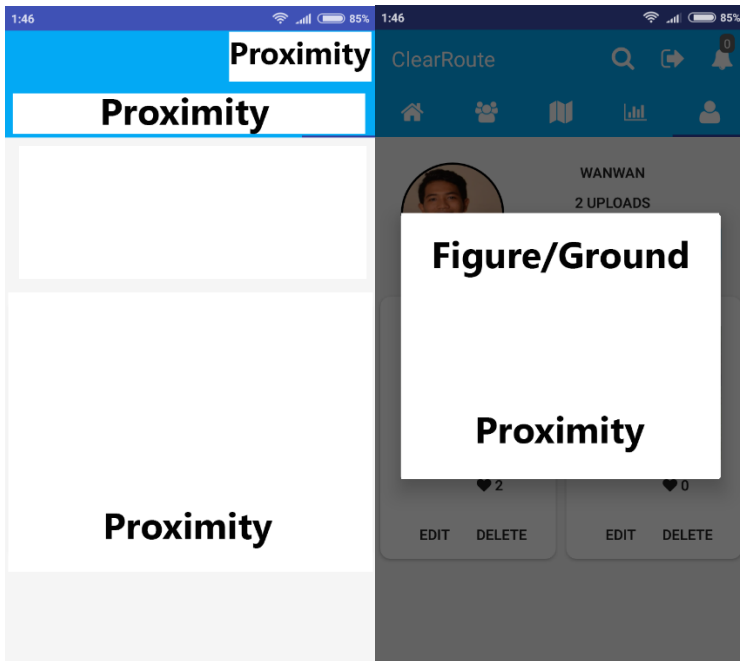
Gambar 3.21 Antarmuka Halaman Peta

Pada halaman peta digunakan dua prinsip *Gestalt*, yaitu *Figure/Ground* dan *Proximity*. *Figure/Ground* digunakan pada saat menampilkan peta dengan isian alamat tujuan. Sedangkan *proximity* digunakan pada pengelompokan *icon toolbar* dan *icon* halaman utama.



Gambar 3.22 Antarmuka Halaman Grafik

Pada halaman grafik digunakan dua prinsip *Gestalt*, yaitu *Common Fate* dan *Proximity*. *Common fate* digunakan pada saat menampilkan animasi grafik dari kondisi cuaca yang ada. Sedangkan *proximity* digunakan pada pengelompokan daftar grafik, *icon toolbar*, dan *icon* halaman utama.



Gambar 3.23 Antarmuka Halaman Profil

Pada halaman profil digunakan dua prinsip *Gestalt*, yaitu *Figure/Ground* dan *Proximity*. *Figure/ground* digunakan pada saat menampilkan pilihan edit. Sedangkan *proximity* digunakan pada pengelompokan daftar berita, *icon toolbar*, *icon* halaman utama, kelompok isian edit, tombol pilihan edit, dan tombol tindakan.

3.6.2.4 Antarmuka Halaman Tambah Berita

Pada antarmuka tersebut terdapat 5 data berita yang perlu diisi pengguna, yaitu judul, deskripsi, lokasi, tanggal, waktu, dan sebuah tombol untuk mengirim data-data tadi ke sistem.



Gambar 3.24 Antarmuka Halaman Tambah Berita

Pada halaman tambah berita, digunakan prinsip *proximity* untuk mengelompokkan sejumlah isian oleh pengguna.

BAB IV

IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak *client* dan *server*, implementasi antarmuka pengguna, dan implementasi proses-proses yang terjadi pada masing-masing kasus penggunaan pada perangkat lunak. Implementasi sistem mengacu pada perancangan yang ditulis pada Bab 3. Namun, tidak menutup kemungkinan adanya perubahan-perubahan dari rancangan tersebut apabila memang diperlukan.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah komputer dan perangkat *smartphone*. Untuk komputer yang digunakan memiliki spesifikasi sebagai berikut.

1. Perangkat keras
 - Prosesor: Intel® Core™ i7-4710HQ CPU @ 2.50GHz.
 - Memori: 8.00 GB.
2. Perangkat lunak
 - Windows 8.1 64 bit sebagai sistem operasi.
 - Android Studio 2.3.2 sebagai IDE untuk implementasi aplikasi *smartphone*.
 - SQLyog Ultimate v8.6 sebagai aplikasi manajemen *database* sistem.
 - Brackets sebagai *editor* untuk merancang *web service* yang digunakan sistem.
 - Power Designer 15.0 untuk merancang *database* dan mendesain diagram perancangan sistem.

Sedangkan perangkat *smartphone* yang digunakan dalam pengembangan dan pengujian sistem yaitu Redmi 4A dengan spesifikasi sebagai berikut.

- Android OS, v6.0 (Marshmallow) .
- CPU Quad-core 1.4 GHz (2GB RAM).
- *WiFi* 802.11 a/b/g/n/ac.

Dan perangkat *smartphone* lainnya yaitu OPPO Find7 dengan spesifikasi sebagai berikut.

- Android OS, v4.3 (Jelly Bean) .
- ColorOS V1.2.9i CPU Quad-core 2.5 GHz (3GB RAM).
- *WiFi* 802.11 a/b/g/n/ac.

4.2 Implementasi Perangkat Lunak

Pada bagian ini akan dijelaskan mengenai implementasi proses yang telah dijabarkan pada perancangan modul di bab III ke dalam sebuah kode. Proses yang dibahas adalah proses-proses yang terjadi dalam sistem sesuai dengan kebutuhan fungsional.

4.2.1 Implementasi Kasus Penggunaan

Dalam subbab ini akan diimplementasikan fungsi-fungsi utama dari aplikasi ClearRoute. Implementasi yang ada berbentuk *pseudocode*.

4.2.1.1 Implementasi Proses Mendaftar Akun

Proses ini dilakukan oleh pengguna untuk mendaftarkan akun dirinya ke dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan empat data pengguna. Data Nama dibutuhkan untuk dijadikan nama pengguna di dalam *Community Based Service* yang mana akan dikenali oleh pengguna yang lain. Data *email* dan *password* dibutuhkan untuk melakukan proses autentifikasi pengguna pada saat *login* nantinya. Data nomor *handphone* dibutuhkan untuk proses pemulihan akun pengguna bilamana pengguna lupa akan *password* akunya.

1	INCLUDE internet
2	INPUT nama, email, password, phone
3	Binding_Objects()
4	Check_Input()
5	Call_Service()
6	Get_Response()
7	SaveSessions()
8	StartActivity()

Gambar 4.1 Pseudocode Mendaftar Akun Pada Aplikasi

1	INCLUDE internet
2	Get_Call_Parameter()
3	Check_Parameter_From_Database()
4	IF Parameter Free
5	INSERT Parameter into Database
6	Send_Response()

Gambar 4.2 Pseudocode Mendaftar Akun Pada Web Service

Fungsi `Binding_Objects()` digunakan untuk mengubah masukan pengguna di XML ke dalam Java. Hal ini dilakukan agar masukan pengguna dapat diproses di dalam *file* Java.

Fungsi `Check_Input()` digunakan untuk memeriksa masukan pengguna sudah sesuai dengan ketentuan, seperti halnya masukan tidak boleh kosong, dan sebagainya.

Fungsi `Call_Service()` dan `Get_Response()` merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan library Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*

Fungsi `SaveSessions()` digunakan untuk menyimpan beberapa data untuk fungsi lainnya di tempat yang berbeda dalam satu aplikasi.

Fungsi `StartActivity()` digunakan untuk menjalankan *activity* Java lainnya yang dituju. Dengan fungsi ini kita dapat berpindah antar *activity*.

Fungsi `Check_Parameter_From_Database()` digunakan untuk melihat apakah nama dan *email* yang didapat sudah dipakai atau belum.

4.2.1.2 Implementasi Proses Melakukan Login

Proses ini dilakukan oleh pengguna untuk melakukan *login* ke dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan dua data pengguna. Data *email* dan *password* dibutuhkan untuk melakukan proses autentifikasi pengguna pada saat *login*. Data nomor *handphone* dibutuhkan untuk proses pemulihan akun pengguna bilamana pengguna lupa akan *password* akunnya ketika akan *login*.

1	INCLUDE internet
2	INPUT email, password
3	IF Session ACTIVE
4	GO TO Home
5	Binding_Objects()
6	Check_Input()
7	Call_Service()
8	Get_Response()
9	SaveSessions()
10	StartActivity()

Gambar 4.3 Pseudocode Melakukan Login pada Aplikasi

1	INCLUDE internet
2	Get_Call_Parameter()
3	Check_Parameter_From_Database()
4	IF Parameter Exist
5	Status True
6	Send_Response()

Gambar 4.4 Pseudocode Melakukan Login pada Web Service

Fungsi `Binding_Objects()` digunakan untuk mengubah masukan pengguna di XML ke dalam Java. Hal ini dilakukan agar masukan pengguna dapat diproses di dalam *file* Java.

Fungsi `Check_Input()` digunakan untuk memeriksa masukan pengguna sudah sesuai dengan ketentuan, seperti halnya masukan tidak boleh kosong, dan sebagainya.

Fungsi `Call_Service()` dan `Get_Response()` merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan library Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*.

Fungsi `SaveSessions()` digunakan untuk menyimpan beberapa data untuk digunakan di fungsi lainnya di tempat yang berbeda dalam satu aplikasi.

Fungsi `StartActivity()` digunakan untuk menjalankan *activity* Java lainnya yang dituju. Dengan fungsi ini kita dapat berpindah antar *activity*.

Fungsi `Check_Parameter_From_Database()` digunakan untuk melihat apakah nama dan *email* yang didapat sudah dipakai atau belum.

4.2.1.3 Implementasi Proses Melihat Profil

Proses ini dilakukan oleh pengguna untuk melihat profil dirinya di dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan data pengguna dalam *session*. Data nama dibutuhkan sebagai parameter untuk mendapatkan data berita dalam basis data.

1	<code>INCLUDE internet</code>
2	<code>Get_Session()</code>
3	<code>Call_Service()</code>
4	<code>Get_Response()</code>

Gambar 4.5 Pseudocode Melihat Profil pada Aplikasi

1	INCLUDE internet
2	Get_Call_Parameter()
3	Get_Data_From_Database()
4	Send_Response()

Gambar 4.6 Pseudocode Melihat Profil pada Web Service

Fungsi `Get_Session()` digunakan untuk mengambil data yang sudah tersimpan dalam *session* yang mana nantinya akan digunakan sebagai parameter pengambilan data dari basis data.

Fungsi `Call_Service()` dan `Get_Response()` merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan *library* Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*.

Fungsi `Get_Data_From_Database()` digunakan untuk mendapatkan data-data yang dibutuhkan untuk ditampilkan di aplikasi.

4.2.1.4 Implementasi Proses Mengubah Profil

Proses ini dilakukan oleh pengguna untuk mengubah profil dirinya di dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan data pengguna dalam *session*. Data nama dibutuhkan sebagai parameter untuk mendapatkan data profil dalam basis data.

1	INCLUDE internet
2	Get_Session()
3	Call_Service()
4	Get_Response()
	IF Edit_Account CLICKED
	Show Dialog
	IF Edit_Profile CLICKED
	Show Dialog
	INPUT nama, email, phone
	Binding_Objects()
	Check_Input()
	Call_Service()

	Get_Response()
--	----------------

Gambar 4.7 Pseudocode Mengubah Profil pada Aplikasi

1	INCLUDE internet
2	Get_Call_Parameter()
3	UPDATE User SET Data into Database
4	Send_Response()

Gambar 4.8 Pseudocode Mengubah Profil pada Web Service

Fungsi Get_Session() digunakan untuk mengambil data yang sudah tersimpan dalam *session* yang mana nantinya akan digunakan sebagai parameter pengambilan data dari basis data.

Fungsi Call_Service() dan Get_Response() merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan *library* Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*.

Fungsi Binding_Objects() digunakan untuk mengubah masukan pengguna di XML ke dalam Java. Hal ini dilakukan agar masukan pengguna dapat diproses di dalam *file* Java.

Fungsi Check_Input() digunakan untuk memeriksa masukan pengguna sudah sesuai dengan ketentuan, seperti halnya masukan tidak boleh kosong, dan sebagainya.

4.2.1.5 Implementasi Proses Melihat Ramalan Terkini

Proses ini dilakukan oleh pengguna untuk melihat ramalan terkini di dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan data lokasi pengguna. Data lokasi dibutuhkan sebagai parameter untuk mendapatkan data ramalan terkini dalam basis data.

1	INCLUDE internet
2	Get_Location()
3	Call_Service()
4	Get_Response()

Gambar 4.9 Pseudocode Melihat Ramalan pada Aplikasi

Fungsi `Get_Location()` digunakan untuk mendapatkan data lokasi pengguna saat ini secara *real time* yang mana akan dikirimkan untuk mendapatkan data-data ramalan, cuaca, grafik, dan sebagainya.

Fungsi `Call_Service()` dan `Get_Response()` merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan *library* Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*.

4.2.1.6 Implementasi Proses Melihat Berita Cuaca

Proses ini dilakukan oleh pengguna untuk melihat berita cuaca dari pengguna lain di dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan data lokasi pengguna. Data lokasi dibutuhkan sebagai parameter untuk mendapatkan data berita cuaca dalam basis data.

1	<code>INCLUDE internet</code>
2	<code>Get_Location()</code>
3	<code>Call_Service()</code>
4	<code>Get_Response()</code>

Gambar 4.10 Pseudocode Melihat Berita pada Aplikasi

1	<code>INCLUDE internet</code>
2	<code>Get_Call_Parameter()</code>
3	<code>Get_Data_From_Database()</code>
4	<code>Send_Response()</code>

Gambar 4.11 Pseudocode Melihat Berita pada Web Service

Fungsi `Get_Location()` digunakan untuk mendapatkan data lokasi pengguna saat ini secara *real time* yang mana akan dikirimkan untuk mendapatkan data-data ramalan, cuaca, grafik, dan sebagainya.

Fungsi `Call_Service()` dan `Get_Response()` merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan *library*

Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*.

Fungsi `Get_Data_From_Database()` digunakan untuk mendapatkan data-data yang dibutuhkan untuk ditampilkan di aplikasi.

4.2.1.7 Implementasi Proses Membuat Berita Cuaca

Proses ini dilakukan oleh pengguna untuk membuat berita cuaca di dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan lima data detail berita. Data judul, deskripsi, lokasi, tanggal, dan waktu merupakan data ciri dari berita yang akan dimasukkan ke dalam basis data.

1	INCLUDE internet
2	INPUT judul, deskripsi, lokasi, tanggal, waktu
3	Binding_Objects()
4	Check_Input()
5	Call_Service()
6	Get_Response()
7	StartActivity()

Gambar 4.12 Pseudocode Membuat Berita pada Aplikasi

1	INCLUDE internet
2	Get_Call_Parameter()
3	INSERT Parameter into Database
4	Send_Response()

Gambar 4.13 Pseudocode Membuat Berita pada Web Service

Fungsi `Binding_Objects()` digunakan untuk mengubah masukan pengguna di XML ke dalam Java. Hal ini dilakukan agar masukan pengguna dapat diproses di dalam *file* Java.

Fungsi `Check_Input()` digunakan untuk memeriksa masukan pengguna sudah sesuai dengan ketentuan, seperti halnya masukan tidak boleh kosong, dan sebagainya.

Fungsi `Call_Service()` dan `Get_Response()` merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan *library* Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*.

Fungsi `StartActivity()` digunakan untuk menjalankan *activity* Java lainnya yang dituju. Dengan fungsi ini kita dapat berpindah antar *activity*.

4.2.1.8 Implementasi Proses Melihat Rekomendasi Rute

Proses ini dilakukan oleh pengguna untuk melihat rekomendasi rute di dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan data lokasi pengguna dan tempat tujuan. Data lokasi dibutuhkan sebagai parameter untuk mendapatkan data rute dalam basis data.

1	INCLUDE internet
2	INPUT tempat_tujuan
3	Binding_Objects()
4	Get_Location()
5	Call_Service()
6	Get_Response()

Gambar 4.14 Pseudocode Melihat Rute pada Aplikasi

Fungsi `Binding_Objects()` digunakan untuk mengubah masukan pengguna di XML ke dalam Java. Hal ini dilakukan agar masukan pengguna dapat diproses di dalam *file* Java.

Fungsi `Get_Location()` digunakan untuk mendapatkan data lokasi pengguna saat ini secara *real time* yang mana akan dikirimkan untuk mendapatkan data-data ramalan, cuaca, grafik, dan sebagainya.

Fungsi `Call_Service()` dan `Get_Response()` merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan *library* Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*.

4.2.1.9 Implementasi Proses Melihat Grafik

Proses ini dilakukan oleh pengguna untuk melihat grafik di dalam aplikasi ClearRoute. Di dalam proses ini dibutuhkan data lokasi pengguna. Data lokasi dibutuhkan sebagai parameter untuk mendapatkan data grafik dalam basis data.

1	INCLUDE internet
2	Get_Location()
3	Call_Service()
4	Get_Response()

Gambar 4.15 Pseudocode Melihat Grafik pada Aplikasi

Fungsi Get_Location() digunakan untuk mendapatkan data lokasi pengguna saat ini secara *real time* yang mana akan dikirimkan untuk mendapatkan data-data ramalan, cuaca, grafik, dan sebagainya.

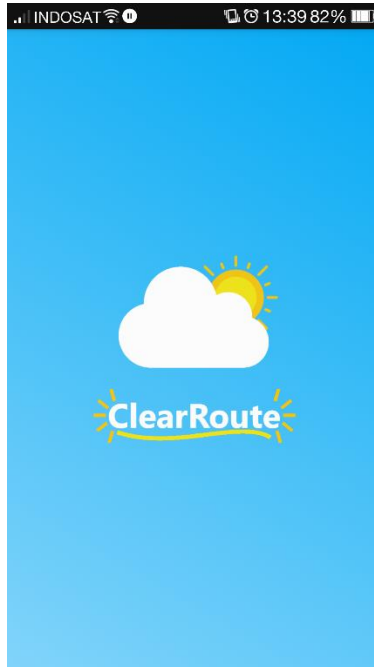
Fungsi Call_Service() dan Get_Response() merupakan satu kesatuan fungsi yang didapatkan dengan menjalankan *library* Retrofit dengan RestAPI. Fungsi ini mengirimkan data-data masukan pengguna ke *web service*.

4.2.2 Implementasi Antarmuka

Pada subbab ini akan dibahas implementasi antarmuka aplikasi berdasarkan rancangan antarmuka yang telah dibahas pada bab 3. Antarmuka yang akan dibahas terdiri dari beberapa bagian yang dijelaskan sebagai berikut.

4.2.2.1 Antarmuka Halaman *Splash Screen*

Halaman *splash screen* merupakan halaman depan saat pengguna pertama kali membuka aplikasi. Halaman ini terdiri dari gambar logo aplikasi.



Gambar 4.16 Antarmuka Halaman *Splash Screen*

Kegunaan halaman *splash screen* adalah untuk mengecek beberapa kondisi mulai dari ketersediaan koneksi internet perangkat, ketersediaan WiFi yang dapat digunakan, serta kesesuaian WiFi yang ditangkap dengan WiFi yang terdapat dalam *database* sistem. Gambar 4.16 menunjukkan implementasi halaman *splash screen*.

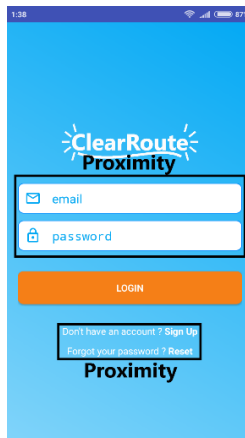
4.2.2.2 Antarmuka Halaman *Login*

Halaman login merupakan halaman pintu utama pengguna untuk menikmati seluruh fitur utama aplikasi ini. Halaman ini terdiri dari logo aplikasi, dua buah `EditText` untuk pengguna memasukkan data dan dua buah *link* menuju halaman lain.



Gambar 4.17 Antarmuka Halaman *Login*

Kegunaan halaman *login* adalah untuk melakukan autentifikasi pengguna yang mengakses aplikasi. Disini pengguna harus memasukkan data *email* dan *password* yang sudah terdaftar pada basis data aplikasi. Pengguna yang tidak berhak atau salah memasukkan data akan mendapatkan notifikasi di layarnya sebagai bentuk peringatan.

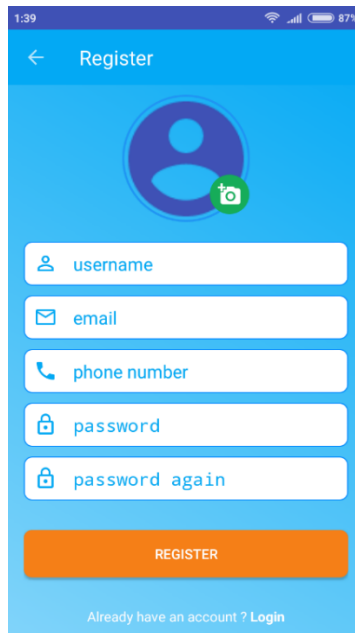


Gambar 4.18 Gestalt Principles Pada Halaman *Login*

Pada halaman login diterapkan satu prinsip *Gestalt* yaitu *Proximity*. Kolom isian untuk pengguna dijadikan dalam satu kelompok dengan komposisi jarak maupun ukuran yang berbeda dengan objek lain didekatnya. Selain itu, juga ada tautan yang dikelompokkan dibawah tombol *login*.

4.2.2.3 Antarmuka Halaman *Register*

Halaman *register* merupakan halaman dimana pengguna memasukkan data pribadinya sebagai ciri akses pengguna terhadap aplikasi. Di dalam halaman ini terdapat sebuah tempat pengguna memasukkan foto dirinya atau gambar lain sebagai ciri pengguna tersebut, lima buah *EditText* untuk pengguna memasukkan data dirinya, dan sebuah *link* menuju halaman *login* kembali.



Gambar 4.19 Antarmuka Halaman *Register*

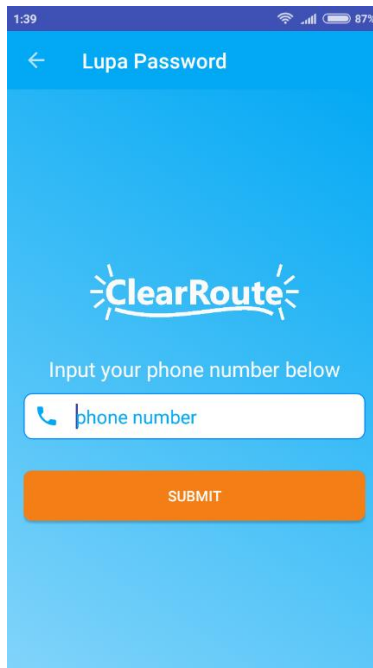
Kegunaan halaman *register* adalah wadah pengguna baru yang tertarik menggunakan fitur-fitur unik aplikasi ini untuk mendaftarkan dirinya dengan akun yang valid. Disini pengguna akan memasukkan data *email* dan *password* yang akan digunakan pada saat *login*, data nomor telepon akan digunakan pengguna pada saat pemulihan akun pengguna, data *username* akan digunakan sebagai nama akun pengguna.

Gambar 4.20 Gestalt Principles Pada Halaman Register

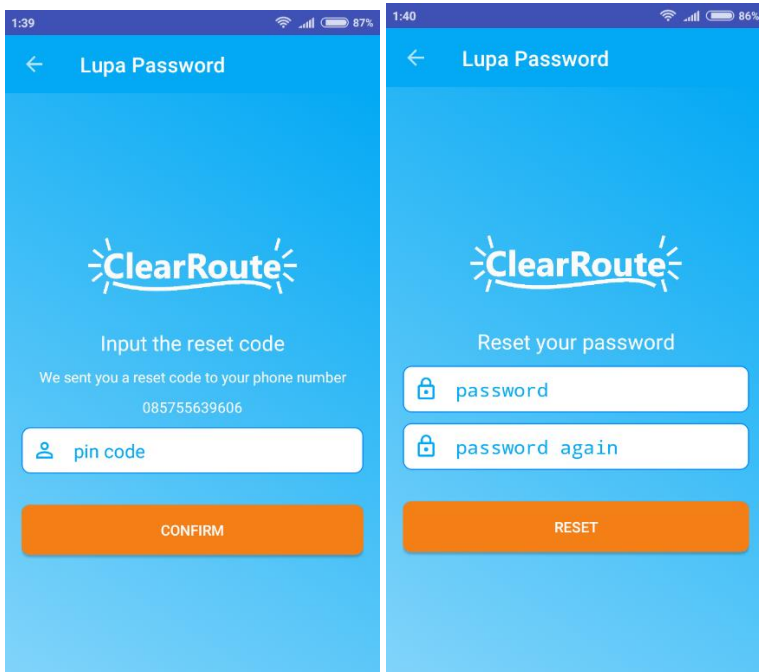
Pada halaman *register* diterapkan satu prinsip *Gestalt* yaitu *Proximity*. Kolom isian untuk pengguna dijadikan dalam satu kelompok dengan komposisi jarak maupun ukuran yang berbeda dengan objek lain didekatnya.

4.2.2.4 Antarmuka Halaman *Forgot Password*

Halaman *forgot password* merupakan halaman dimana pengguna bisa gunakan untuk memulihkan akun pengguna ketika pengguna tidak bisa mengakses akun dikarenakan salah atau lupa *password*. Di dalam halaman ini dibutuhkan nomor telepon untuk dikirimkan data kode pin yang diperlukan untuk dipulihkan. Tentunya nomor telepon yang dimasukkan harus sesuai dengan nomor telepon yang terdaftar.

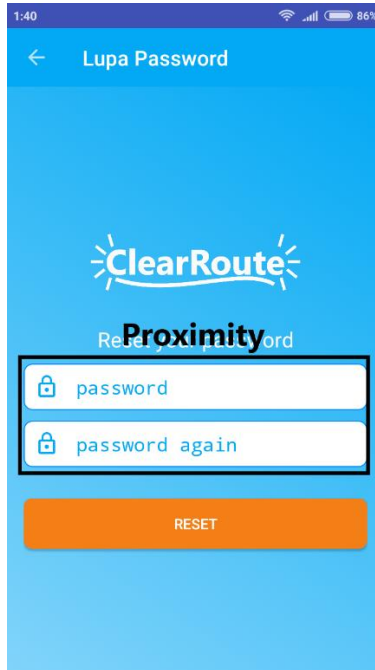


The screenshot shows a mobile application interface for the 'Forgot Password' screen. At the top, there is a status bar with the time '1:39', signal strength, and battery level at '87%'. Below the status bar is a blue header with a back arrow and the text 'Lupa Password'. The main background is a light blue gradient. In the center, the 'ClearRoute' logo is displayed, featuring the word 'ClearRoute' in white with a stylized sunburst icon. Below the logo, the text 'Input your phone number below' is shown. Underneath this text is a white input field with a blue telephone icon on the left and the placeholder text 'phone number'. At the bottom of the form is a large orange button with the word 'SUBMIT' in white capital letters.



Gambar 4.21 Antarmuka Halaman *Forgot Password*

Kegunaan halaman *forgot password* adalah sebagai wadah pengguna untuk memulihkan akun dan dapat menikmati kembali fitur-fitur dari aplikasi. Disini pengguna akan memasukkan nomor telepon, kode pin, dan *password* barunya jika berjalan dengan lancar. Jika ada kesalahan, maka aplikasi akan memberikan peringatan kepada pengguna berupa *snackbar* di bagian bawah aplikasi.

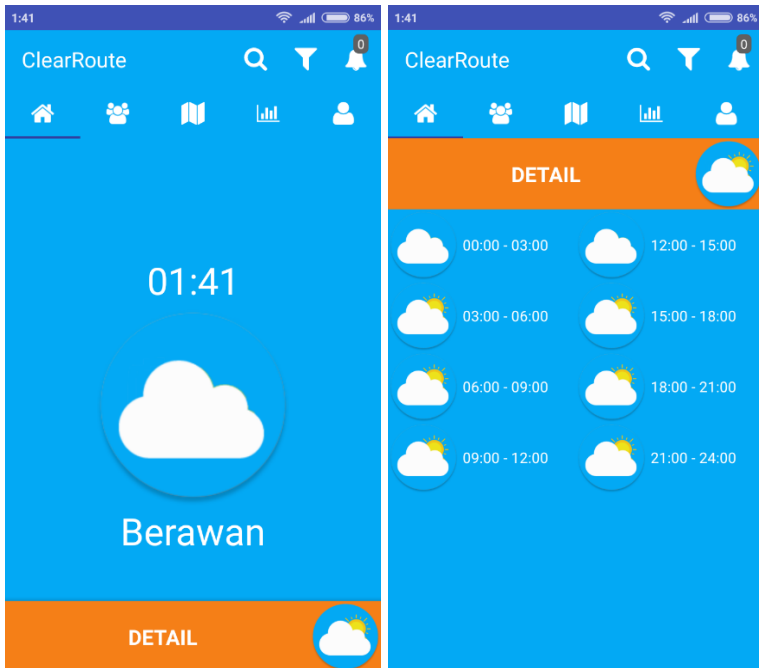


Gambar 4.22 Gestalt Principles Pada Halaman *Forgot Password*

Pada halaman *forgot password* diterapkan satu prinsip *Gestalt* yaitu *Proximity*. Kolom isian untuk pengguna dijadikan dalam satu kelompok dengan komposisi jarak maupun ukuran yang berbeda dengan objek lain didekatnya.

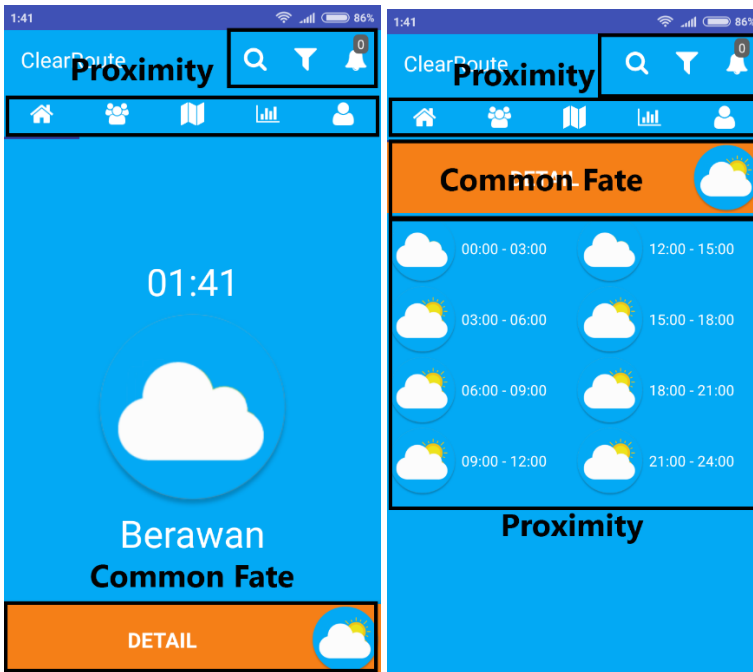
4.2.2.5 Antarmuka Halaman Ramalan

Halaman ramalan merupakan halaman dimana pengguna dapat melihat kondisi cuaca saat ini di lokasi dia berada dan kondisi cuaca selama satu hari penuh. Di dalam halaman ini terdapat tampilan waktu yang menunjukkan waktu cuaca diambil, tampilan kondisi cuaca beserta *icon* yang mencerminkan kondisi cuaca tersebut.



Gambar 4.23 Antarmuka Halaman Ramalan

Kegunaan halaman ramalan adalah untuk menunjukkan pada pengguna kondisi cuaca dia saat ini berada. Di dalam halaman ini pengguna dapat menggeser tanda 'Detail' ataupun menekannya untuk melihat detail cuaca dalam satu hari. Jika kondisi cuacanya tidak sesuai dengan keadaan disekitar pengguna, pengguna dapat memberikan pembaharuan terkini di halaman tambah berita.



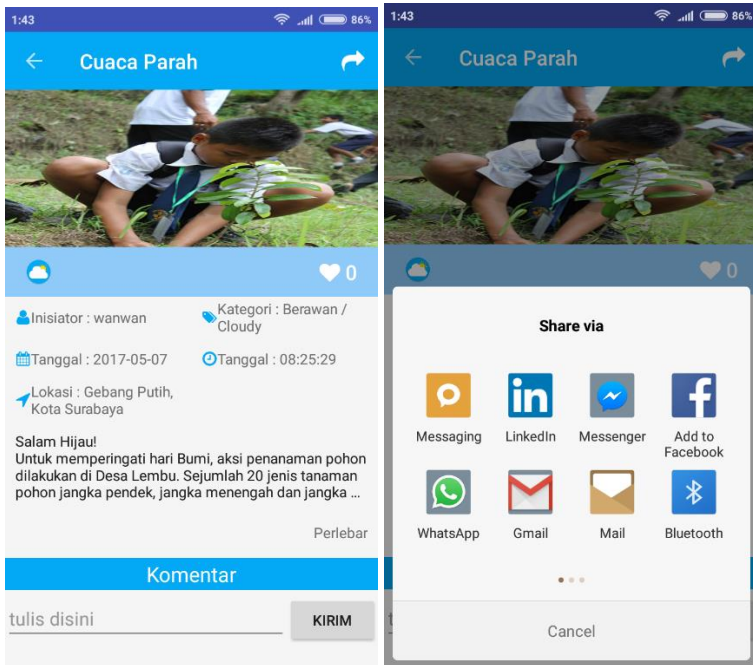
Gambar 4.24 Gestalt Principles Pada Halaman Ramalan

Pada halaman ramalan diterapkan dua prinsip *Gestalt* yaitu *Proximity* dan *Common Fate*. Pada halaman bagian atas dikelompokkan menjadi satu kumpulan menu *toolbar* yang bisa digunakan oleh pengguna. Di bawah *toolbar* terdapat *view pager* atau kumpulan halaman yang ditunjukkan dalam bentuk *icon* unik dan dikelompokkan dengan jarak dan komposisi yang sama. Untuk detail cuaca dengan batas setiap tiga jam juga dikelompokkan. Untuk halaman ini penggunaan *Common Fate* memanfaatkan *Sliding Panel* agar layar terkesan panjang dibawah tanpa adanya pengguna melakukan *scroll*. Selain bisa digeser keatas, tombol detail juga bisa ditekan oleh pengguna sehingga halaman akan menggerakkan dirinya keatas maupun kebawah.

4.2.2.6 Antarmuka Halaman Berita

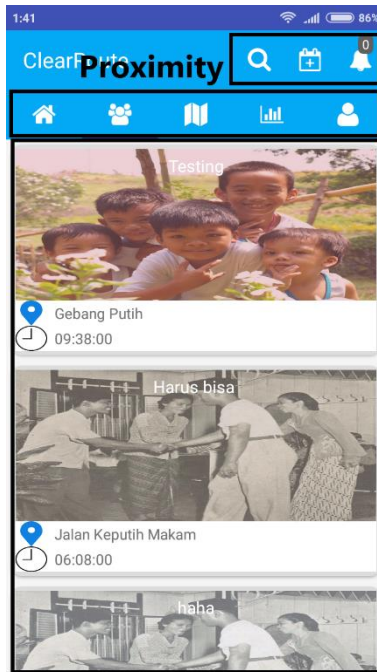
Halaman berita merupakan halaman dimana pengguna dapat melihat berita-berita cuaca di wilayahnya yang mana berita-berita tersebut hasil laporan dari pengguna lain. Di halaman ini terdapat tampilan daftar berita dengan komposisi judul, lokasi, waktu, dan foto berita.





Gambar 4.25 Antarmuka Halaman Berita

Kegunaan dari halaman berita adalah pengguna dapat melihat berita-berita cuaca tidak terduga yang terjadi dan tidak sesuai dengan ramalan cuaca yang diberikan. Disini pengguna dapat melihat rincian berita berupa akun *inisiator*, kategori, Tanggal, waktu, lokasi, dan deskripsi berita. Di halaman ini pengguna juga bisa memberikan komentar terhadap berita yang dia lihat. Di halaman ini juga pengguna bisa membagikan berita melalui media sosial lainnya agar orang-orang terdekat dengan kita lebih waspada terhadap kondisi cuaca yang dilaporkan.



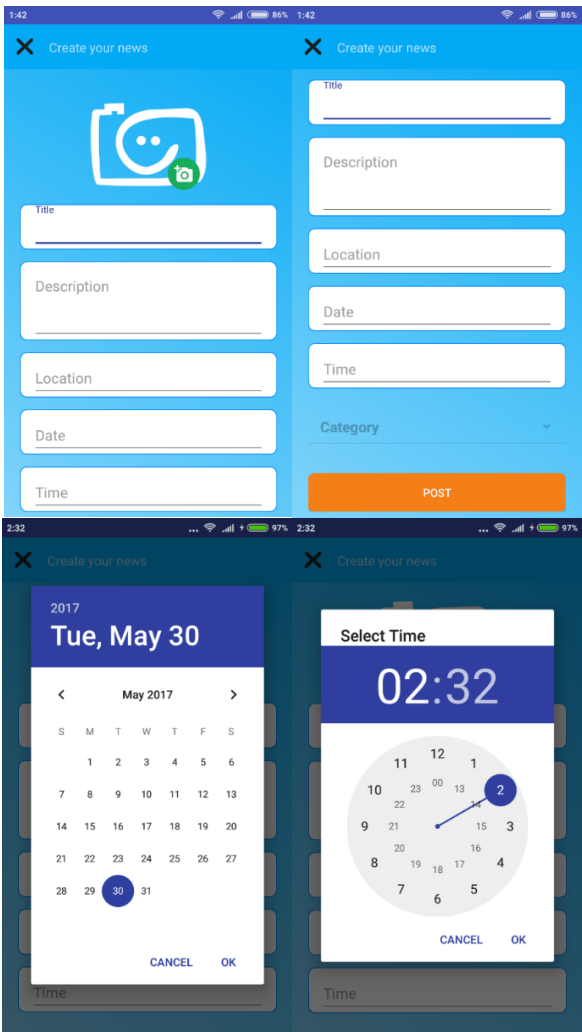
Gambar 4.26 Gestalt Principles Pada Halaman Berita

Pada halaman berita diterapkan satu prinsip *Gestalt* yaitu *Proximity*. Pada halaman bagian atas dikelompokkan menjadi satu kumpulan menu *toolbar* yang bisa digunakan oleh pengguna. Di bawah *toolbar* terdapat *view pager* atau kumpulan halaman yang ditunjukkan dalam bentuk *icon* unik dan dikelompokkan dengan jarak dan komposisi yang sama. Di bagian utama halaman juga dikelompokkan daftar berita secara vertikal untuk dilihat oleh pengguna.

4.2.2.7 Antarmuka Halaman Tambah Berita

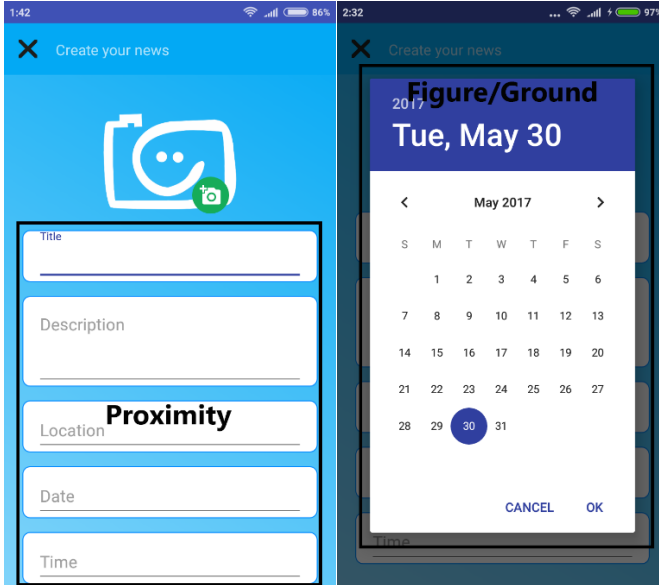
Halaman tambah berita merupakan halaman dimana pengguna dapat menambahkan berita cuaca terkini di lingkungan sekitarnya. Di halaman ini terdapat sebuah tampilan untuk

pengguna memasukkan foto kondisi cuaca tersebut, lima buah masukan berisi detail berita cuaca tersebut.



Gambar 4.27 Antarmuka Halaman Tambah Berita

Kegunaan dari halaman tambah berita adalah untuk menambahkan berita terbaru sehingga dapat diketahui dan menjadi kewaspadaan banyak orang. Di dalam halaman ini pengguna dapat memasukkan data tanggal dan waktu dengan tampilan yang menarik.

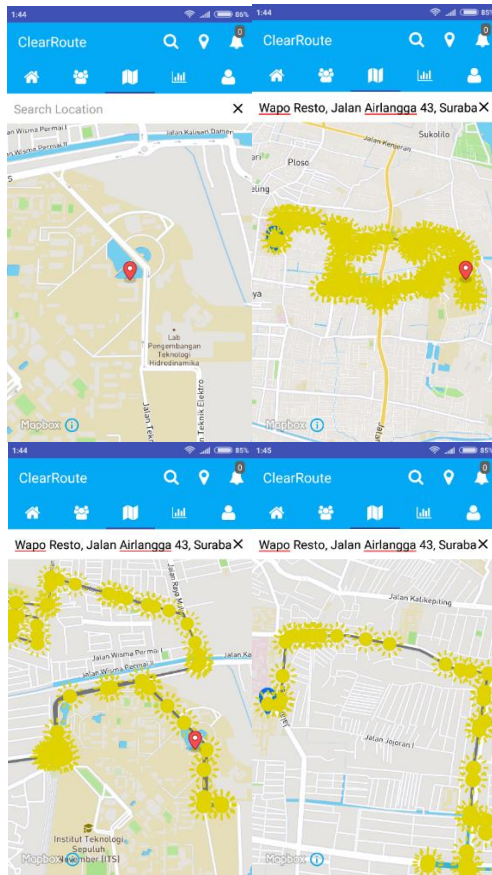


Gambar 4.28 Gestalt Principles Pada Halaman Tambah Berita

Pada halaman tambah berita diterapkan dua prinsip *Gestalt* yaitu *Proximity* dan *Figure/Ground*. Kolom isian untuk pengguna dijadikan dalam satu kelompok dengan komposisi jarak maupun ukuran yang berbeda dengan objek lain didekatnya. Bentuk isian pada tanggal dan jam yang seakan-akan menunjukkan adanya perpaduan antara dua objek sehingga terlihat adanya *foreground* dan *background*.

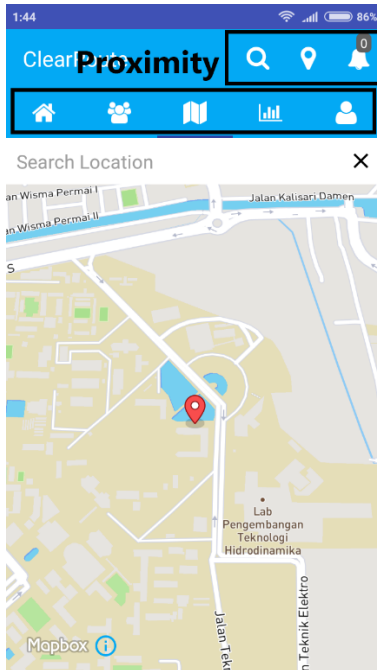
4.2.2.8 Antarmuka Halaman Peta

Halaman peta merupakan halaman dimana pengguna dapat mengetahui rekomendasi rute perjalanan ke tempat tujuan dengan minimal terkena hujan. Di halaman ini terdapat tampilan masukan untuk diisi pengguna berupa lokasi tujuan, tampilan peta dengan komposisi *marker* dirinya, *marker* tujuan, *marker* cuaca, dan jalur berupa *polyline*.



Gambar 4.29 Antarmuka Halaman Peta

Kegunaan dari halaman peta adalah pengguna dapat memilih rute yang mana minimal terkena hujan. Di halaman ini disajikan tiga rekomendasi rute yang bisa dipilih oleh pengguna untuk dilewati.

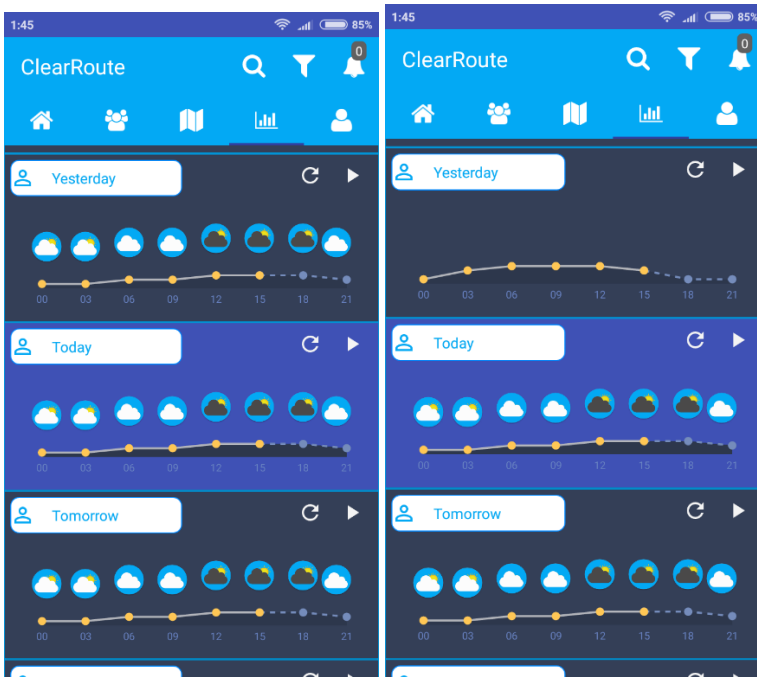


Gambar 4.30 Gestalt Principles Pada Halaman Peta

Pada halaman peta diterapkan satu prinsip *Gestalt* yaitu *Proximity*. Pada halaman bagian atas dikelompokkan menjadi satu kumpulan menu *toolbar* yang bisa digunakan oleh pengguna. Di bawah *toolbar* terdapat *view pager* atau kumpulan halaman yang ditunjukkan dalam bentuk *icon* unik dan dikelompokkan dengan jarak dan komposisi yang sama.

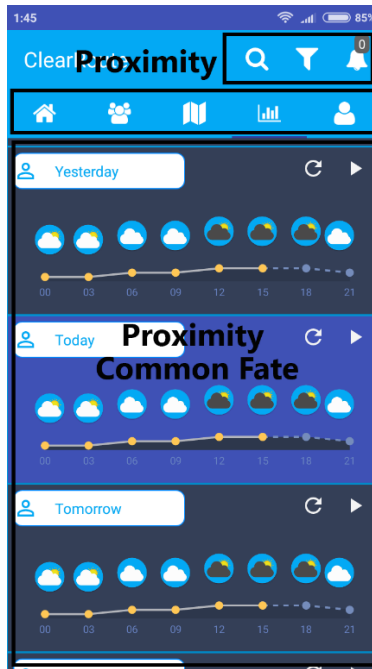
4.2.2.9 Antarmuka Halaman Grafik

Halaman grafik merupakan halaman dimana pengguna dapat melihat grafik cuaca harian dalam rentang waktu tiga hari sebelum hingga tiga hari setelahnya. Di dalam halaman ini terdapat daftar grafik cuaca harian, *icon* cuaca yang menggambarkan keadaan cuaca di dalam grafik.



Gambar 4.31 Antarmuka Halaman Grafik

Kegunaan halaman grafik adalah wadah bagi pengguna dapat memprediksikan kondisi cuaca di lain hari dengan menggunakan data masa lalu, masa sekarang, dan masa depan. Di dalam halaman ini terdapat animasi yang menggerakkan grafik sehingga lebih nyaman digunakan oleh pengguna.

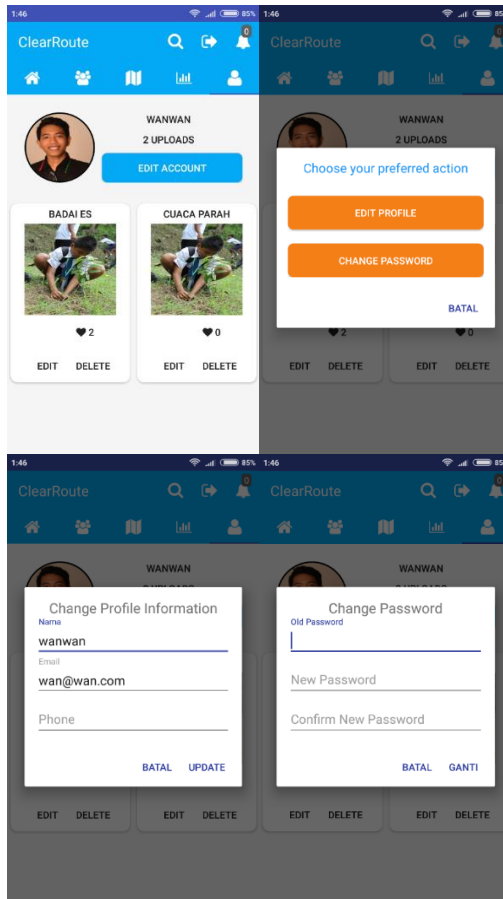


Gambar 4.32 Gestalt Principles Pada Halaman Grafik

Pada halaman grafik diterapkan dua prinsip *Gestalt* yaitu *Proximity* dan *Common Fate*. Pada halaman bagian atas dikelompokkan menjadi satu kumpulan menu *toolbar* yang bisa digunakan oleh pengguna. Di bawah *toolbar* terdapat *view pager* atau kumpulan halaman yang ditunjukkan dalam bentuk *icon* unik dan dikelompokkan dengan jarak dan komposisi yang sama. Daftar grafik yang dikelompokkan secara vertikal dari grafik 3 hari sebelum hingga 3 hari sesudah. Bentuk animasi yang ditunjukkan masing-masing grafik ketika menampilkan bentuk grafiknya pada saat ditekan tombol mulai.

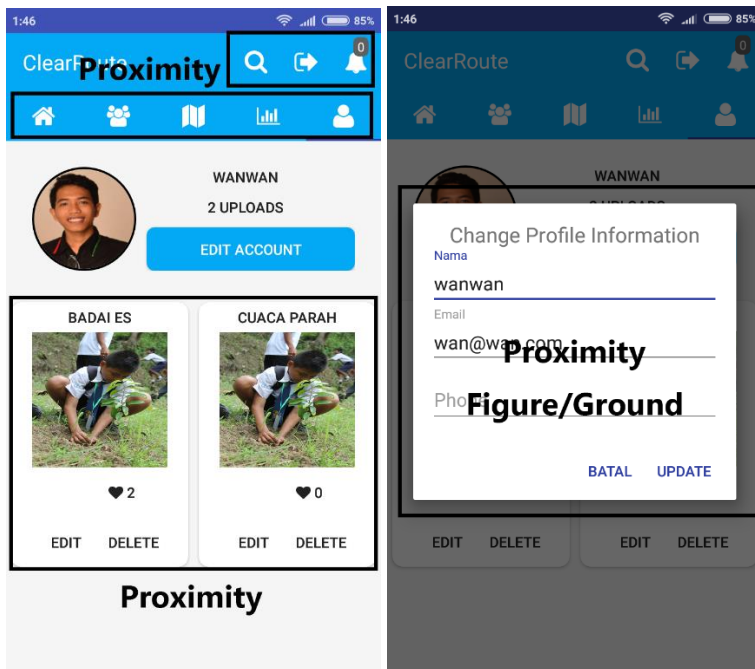
4.2.2.10 Antarmuka Halaman Profil

Halaman profil merupakan halaman dimana pengguna dapat melakukan beberapa hal pada akun miliknya, dari melihat hingga mengubahnya. Di dalam halaman ini ditampilkan foto, nama, banyaknya pembuatan berita beserta daftar berita yang sudah dilaporkan.



Gambar 4.33 Antarmuka Halaman Profil

Kegunaan halaman profil adalah wadah bagi pengguna mendokumentasikan setiap berita yang sudah dia unggah dan dapat dia lihat kembali. Di dalam halaman ini juga disediakan tombol *edit account* untuk mengubah data akun pengguna seperti data pribadi maupun *password*. Selain itu juga terdapat tombol *edit* dan *delete* pada berita dengan tujuan untuk mengubah maupun menghapus berita.



Gambar 4.34 Gestalt Principles Pada Halaman Profil

Pada halaman profil diterapkan dua prinsip *Gestalt* yaitu *Proximity* dan *Figure/Ground*. Pada halaman bagian atas dikelompokkan menjadi satu kumpulan menu *toolbar* yang bisa digunakan oleh pengguna. Di bawah *toolbar* terdapat *view pager* atau kumpulan halaman yang ditunjukkan dalam bentuk *icon* unik

dan dikelompokkan dengan jarak dan komposisi yang sama. Seluruh berita yang pernah diunggah oleh pengguna juga dikelompokkan tersendiri secara vertikal dengan masing-masing baris terdapat dua buah berita. Kolom isian untuk pengguna dijadikan dalam satu kelompok dengan komposisi jarak maupun ukuran yang berbeda dengan objek lain didekatnya. Bentuk isian yang dibuat seakan-akan adanya perpaduan antara dua objek sehingga terlihat seperti adanya *foreground* dan *background*.

BAB V

UJI COBA DAN EVALUASI

Bab ini akan menjelaskan mengenai skenario uji coba pada perangkat lunak yang telah dibangun. Setelah itu, hasil uji coba akan dievaluasi kinerjanya sehingga dapat diputuskan apakah perangkat lunak ini mampu menyelesaikan permasalahan yang telah dirumuskan diawal. Secara garis besar, bab ini berisikan pembahasan mengenai lingkungan pengujian, dan uji kinerja.

5.1 Lingkungan Uji Coba

Lingkungan pengujian merupakan perangkat-perangkat yang dilibatkan dalam proses pengujian. Lingkungan pengujian ini menggunakan perangkat keras berupa *smartphone* berbasis Android yang terdapat fasilitas perangkat *WiFi* serta dapat terhubung dengan internet. Adapun spesifikasi *smartphone* yang digunakan dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Pengujian

Spesifikasi	Keterangan
Nama <i>Smartphone</i>	Redmi 4a
Sistem Operasi	Android OS, v6.0
Prosesor	Quad-core 1.4 GHz
RAM	2 GB
WLAN	Wi-Fi 802.11 a/b/g/n/ac

5.2 Skenario Uji Coba

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan pada aplikasi android ClearRoute. Pengujian yang dilakukan adalah pengujian fungsionalitas dan *heuristic evaluation*. Pengujian didokumentasikan secara sistematis sebagai

tolok ukur keberhasilan sistem. Pengujian dilakukan oleh pengguna biasa (pengguna pada umumnya) dan pengguna ahli (pengguna yang memahami ilmu desain komunikasi visual). Pengguna yang akan melakukan pengujian terhadap aplikasi android ClearRoute yang dibangun terdiri dari 10 pengguna biasa dan 1 pengguna ahli.

5.2.1 Pengujian *Heuristic*

Metode *heuristic evaluation usability* digunakan untuk menghasilkan temuan masalah. Metode ini memungkinkan *evaluator* untuk menemukan kemungkinan masalah *usability* dalam suatu produk atau aplikasi. Saat melakukan evaluasi heuristik, *evaluator* membandingkan seperangkat prinsip kegunaan khusus yang telah ditentukan sebelumnya dengan antarmuka produk saat mencoba menyelesaikan tugas sistem yang sebenarnya. *Evaluator* dapat bekerja secara individu, menggabungkan temuan di kemudian hari, atau mereka mungkin melakukan evaluasi bersamaan dengan masing-masing individu yang berfokus pada beberapa heuristik yang berbeda.

5.2.1.1 Poin *Heuristic*

Ada sembilan poin *heuristic* yang digunakan dalam pengujian ini. Kesembilan poin tersebut ada dalam tabel berikut.

Tabel 5.2 Poin *Heuristic*

Nomor	Poin Heuristik
1	Estetika dan desain yang minimalis
2	Menu atau struktur yang efektif
3	Penggunaan bahasa yang sederhana dan efektif
4	Meminimalisir penggunaan memori
5	Menjaga konsistensi
6	Menyediakan timbal balik
7	Menyediakan jalan keluar

8	Manajemen <i>error</i> secara positif
9	Menyediakan bantuan

5.2.1.2 Prioritas Masalah

Ada dua hal mendasar sebagai parameter dari pengujian ini ialah seberapa besar masalah dari desain aplikasi tersebut dan seberapa mudah memperbaiki masalah tersebut. Parameter-parameter tersebut dapat dilihat di tabel berikut.

Tabel 5.3 Tingkat Gangguan

Tingkat Gangguan	
Tingkat	Definisi
0	Tidak mengganggu penggunaan
1	Kemungkinan gangguan penggunaan sangat kecil
2	Kemungkinan gangguan penggunaan kecil
3	Kemungkinan gangguan penggunaan besar
4	Kemungkinan gangguan penggunaan sangat besar

Tabel 5.4 Tingkat Perbaikan

Tingkat Perbaikan	
Tingkat	Definisi
0	Permasalahan sangat mudah untuk diselesaikan
1	Permasalahan mudah untuk diselesaikan
2	Permasalahan sulit untuk diselesaikan
3	Permasalahan sangat sulit untuk diselesaikan

5.2.2 Pengujian Fungsionalitas

Pengujian fungsionalitas adalah pengujian kebutuhan fungsional yang dilakukan dengan menggunakan metode *black box*. Metode *black box* merupakan metode dimana pengujian ditekankan pada pola masukan dan keluaran yang sesuai dengan

skenario. Pengujian dilakukan dengan mengacu pada kasus penggunaan yang dijelaskan pada Bab 3.

5.2.2.1 Pengujian Mendaftar Akun

Pada pengujian ini pengguna akan mendaftar akun pada aplikasi ClearRoute. Skenario pengujian mendaftar akun dapat dilihat pada Tabel 5.5 dan hasil pengujian pada Gambar 5.1.

Tabel 5.5 Skenario Pengujian Mendaftar Akun

ID	UJ-001
Kasus Penggunaan	Mendaftar Akun
Sub Kasus	-
Nama	Pengujian mendaftar akun
Tujuan Pengujian	Menguji fitur mendaftar akun
Skenario	Pengguna memasukkan data pada halaman register
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tulisan <i>sign up</i> 2. Pengguna mengisi data diri 3. Pengguna menekan tombol <i>submit</i>
Hasil yang Diharapkan	Akun pengguna baru tervalidasi
Hasil yang Didapat	Akun pengguna baru tervalidasi
Hasil Pengujian	Berhasil



Gambar 5.1 Hasil Pengujian Mendaftar Akun

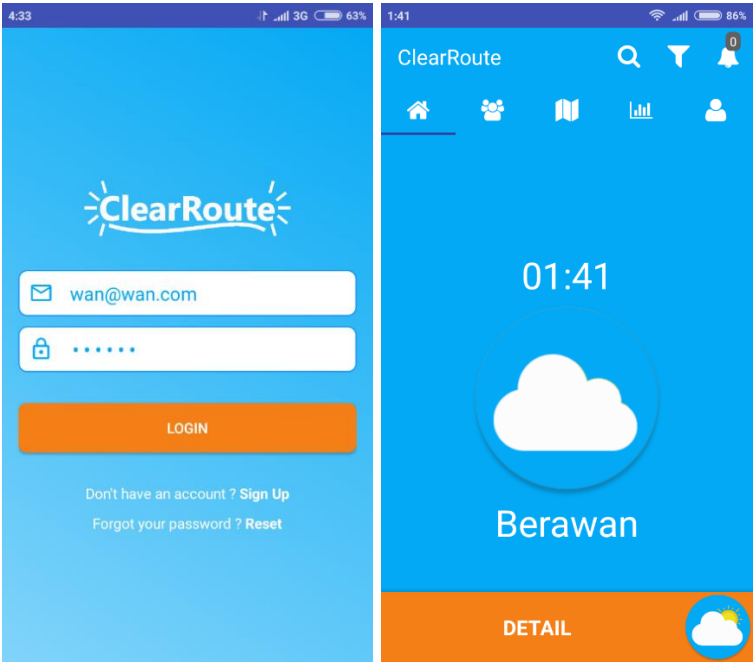
5.2.2.2 Pengujian Melakukan *Login*

Pada pengujian ini pengguna akan melakukan *login* pada aplikasi ClearRoute. Skenario pengujian melakukan *login* dapat dilihat pada Tabel 5.6 dan hasil pengujian pada Gambar 5.3.

Tabel 5.6 Skenario Pengujian Melakukan *Login*

ID	UJ-002
Kasus Penggunaan	Melakukan <i>login</i>
Sub Kasus	-
Nama	Pengujian melakukan <i>login</i>
Tujuan Pengujian	Menguji fitur melakukan <i>login</i>

Skenario	Pengguna memasukkan data pada halaman <i>login</i>
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	1. Pengguna mengisi data diri 2. Pengguna menekan tombol <i>submit</i>
Hasil yang Diharapkan	Pengguna masuk ke dalam halaman <i>Home</i>
Hasil yang Didapat	Pengguna berada di halaman <i>Home</i>
Hasil Pengujian	Berhasil



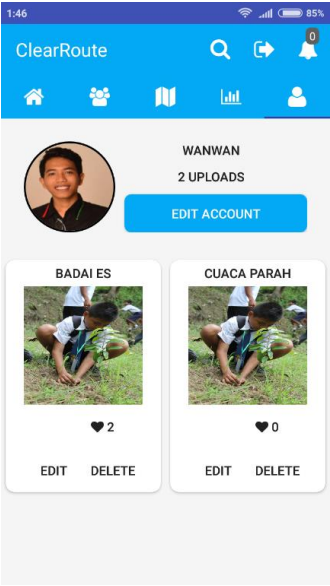
Gambar 5.2 Hasil Pengujian Melakukan Login

5.2.2.3 Pengujian Melihat Profil

Pada pengujian ini pengguna akan melihat profil pada aplikasi ClearRoute. Skenario pengujian melihat profil dapat dilihat pada Tabel 5.7 dan hasil pengujian pada Gambar 5.3.

Tabel 5.7 Skenario Pengujian Melihat Profil

ID	UJ-003
Kasus Penggunaan	Melihat Profil
Sub Kasus	-
Nama	Pengujian melihat profil
Tujuan Pengujian	Menguji fitur melihat profil
Skenario	Pengguna melakukan <i>login</i> hingga menemukan halaman profil
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengisi data diri 2. Pengguna menekan tombol <i>submit</i> 3. Pengguna menggeser layar
Hasil yang Diharapkan	Pengguna berada di halaman profil
Hasil yang Didapat	Pengguna berada di halaman profil
Hasil Pengujian	Berhasil



Gambar 5.3 Hasil Pengujian Melihat Profil

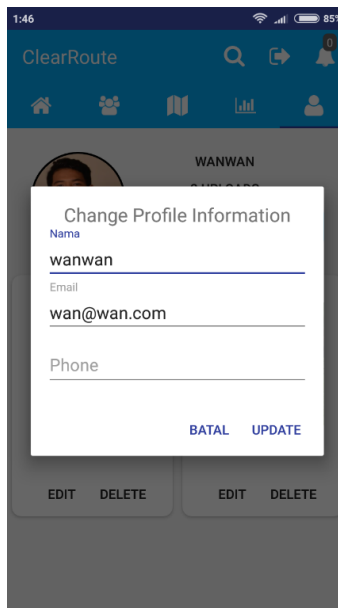
5.2.2.4 Pengujian Mengubah Profil

Pada pengujian ini pengguna akan mengubah profil pada aplikasi ClearRoute. Skenario pengujian mengubah profil dapat dilihat pada Tabel 5.8 dan hasil pengujian pada Gambar 5.4.

Tabel 5.8 Skenario Pengujian Mengubah Profil

ID	UJ-004
Kasus Penggunaan	Mengubah profil
Sub Kasus	-
Nama	Pengujian mengubah profil
Tujuan Pengujian	Menguji fitur mengubah profil
Skenario	Pengguna melakukan login hingga dapat mengubah profil miliknya

Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengisi data diri 2. Pengguna menekan tombol <i>submit</i> 3. Pengguna menggeser layar 4. Pengguna menekan tombol edit pada halaman profil 5. Pengguna mengisi data baru 6. Pengguna menekan tombol <i>update</i>
Hasil yang Diharapkan	Data pengguna berubah sesuai data baru
Hasil yang Didapat	Data pengguna berubah
Hasil Pengujian	Berhasil



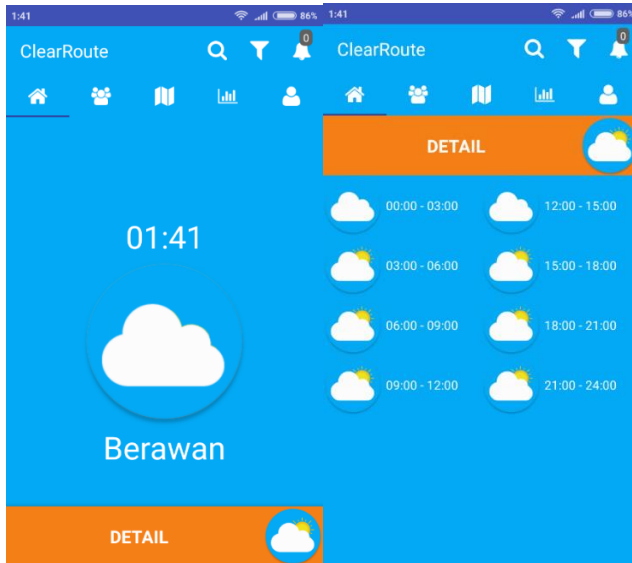
Gambar 5.4 Hasil Pengujian Mengubah Profil

5.2.2.5 Pengujian Melihat Ramalan Terkini

Pada pengujian ini pengguna akan melihat ramalan terkini pada aplikasi ClearRoute. Skenario pengujian melihat ramalan terkini dapat dilihat pada Tabel 5.9 dan hasil pengujian pada Gambar 5.5.

Tabel 5.9 Skenario Pengujian Melihat Ramalan Terkini

ID	UJ-005
Kasus Penggunaan	Melihat Ramalan Terkini
Sub Kasus	-
Nama	Pengujian melihat ramalan terkini
Tujuan Pengujian	Menguji fitur melihat ramalan terkini
Skenario	Pengguna memasukkan data pada halaman <i>login</i>
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengisi data diri 2. Pengguna menekan tombol <i>submit</i> 3. Pengguna menekan tombol <i>details</i>
Hasil yang Diharapkan	Pengguna masuk ke dalam halaman <i>home</i>
Hasil yang Didapat	Pengguna berada di halaman <i>home</i>
Hasil Pengujian	Berhasil



Gambar 5.5 Hasil Pengujian Melihat Ramalan Terkini

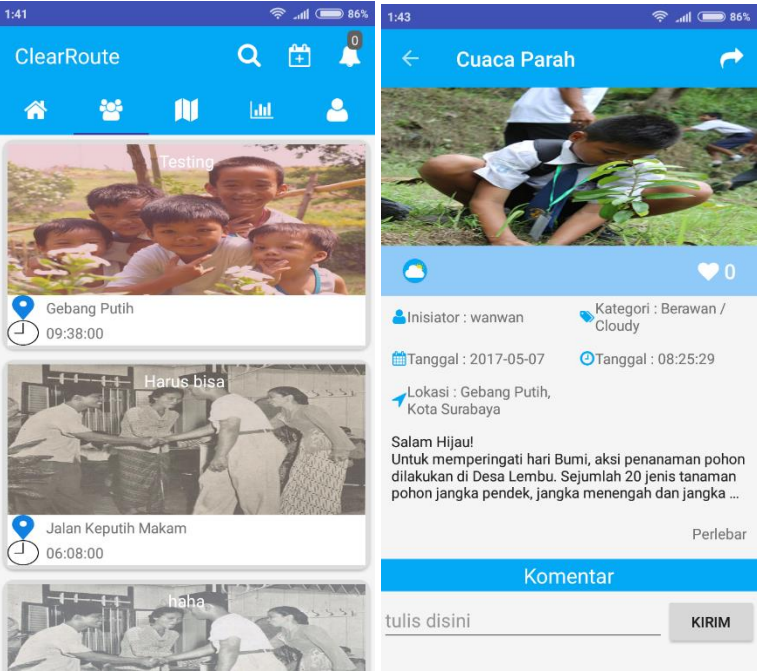
5.2.2.6 Pengujian Melihat Berita Cuaca

Pada pengujian ini pengguna akan melihat berita cuaca pada aplikasi ClearRoute. Skenario pengujian melihat berita cuaca dapat dilihat pada Tabel 5.10 dan hasil pengujian pada Gambar 5.6.

Tabel 5.10 Skenario Pengujian Melihat Berita Cuaca

ID	UJ-006
Kasus Penggunaan	Melihat Berita Cuaca
Sub Kasus	-
Nama	Pengujian melihat berita cuaca
Tujuan Pengujian	Menguji fitur melihat berita cuaca
Skenario	Pengguna melakukan <i>login</i> hingga menemukan halaman berita

Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	1. Pengguna mengisi data diri 2. Pengguna menekan tombol <i>submit</i> 3. Pengguna menggeser layar
Hasil yang Diharapkan	Pengguna berada di halaman berita
Hasil yang Didapat	Pengguna berada di halaman berita
Hasil Pengujian	Berhasil



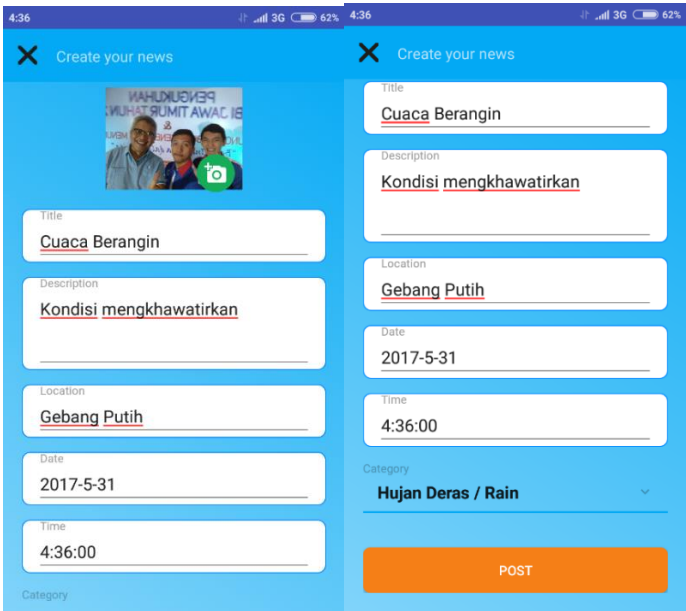
Gambar 5.6 Hasil Pengujian Melihat Berita Cuaca

5.2.2.7 Pengujian Membuat Berita Cuaca

Pada pengujian ini pengguna akan membuat berita cuaca pada aplikasi ClearRoute. Skenario pengujian membuat berita cuaca dapat dilihat pada Tabel 5.11 dan hasil pengujian pada Gambar 5.7.

Tabel 5.11 Skenario Pengujian Membuat Berita Cuaca

ID	UJ-007
Kasus Penggunaan	Membuat Berita Cuaca
Sub Kasus	-
Nama	Pengujian membuat berita cuaca
Tujuan Pengujian	Menguji fitur membuat berita cuaca
Skenario	Pengguna melakukan <i>login</i> hingga berhasil membuat berita
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengisi data diri 2. Pengguna menekan tombol <i>submit</i> 3. Pengguna menggeser layar 4. Pengguna menekan tombol tambah berita 5. Pengguna mengisi data berita 6. Pengguna menekan tombol <i>post</i>
Hasil yang Diharapkan	Pengguna berhasil menambah berita
Hasil yang Didapat	Pengguna berhasil menambah berita
Hasil Pengujian	Berhasil



Gambar 5.7 Hasil Pengujian Membuat Berita Cuaca

5.2.2.8 Pengujian Melihat Rekomendasi Rute

Pada pengujian ini pengguna akan melihat rekomendasi rute pada aplikasi ClearRoute. Skenario pengujian melihat rekomendasi rute dapat dilihat pada Tabel 5.12 dan hasil pengujian pada Gambar 5.8.

Tabel 5.12 Skenario Pengujian Melihat Rekomendasi Rute

ID	UJ-008
Kasus Penggunaan	Melihat Rekomendasi Rute
Sub Kasus	-
Nama	Pengujian melihat rekomendasi rute
Tujuan Pengujian	Menguji fitur melihat rekomendasi rute

Skenario	Pengguna melakukan <i>login</i> hingga menemukan halaman peta
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengisi data diri 2. Pengguna menekan tombol <i>submit</i> 3. Pengguna menggeser layar 4. Pengguna mengisi data alamat tujuan
Hasil yang Diharapkan	Pengguna berada di halaman peta
Hasil yang Didapat	Pengguna berada di halaman peta
Hasil Pengujian	Berhasil



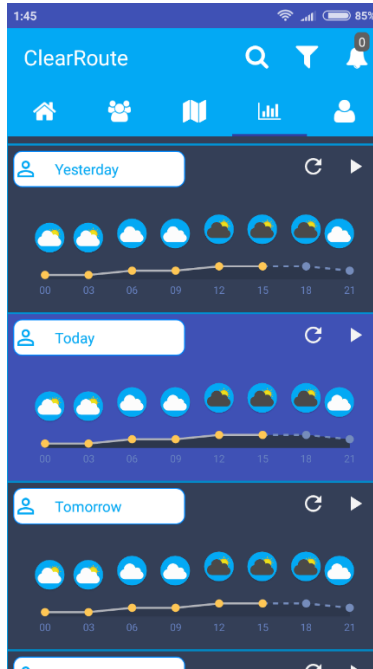
Gambar 5.8 Hasil Pengujian Melihat Rekomendasi Rute

5.2.2.9 Pengujian Melihat Grafik

Pada pengujian ini pengguna akan melihat grafik pada aplikasi ClearRoute. Skenario pengujian melihat grafik dapat dilihat pada Tabel 5.13 dan hasil pengujian pada Gambar 5.9.

Tabel 5.13 Skenario Pengujian Melihat Grafik

ID	UJ-009
Kasus Penggunaan	Melihat Grafik
Sub Kasus	-
Nama	Pengujian melihat grafik
Tujuan Pengujian	Menguji fitur melihat grafik
Skenario	Pengguna melakukan <i>login</i> hingga menemukan halaman grafik
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengisi data diri 2. Pengguna menekan tombol <i>submit</i> 3. Pengguna menggeser layar
Hasil yang Diharapkan	Pengguna berada di halaman grafik
Hasil yang Didapat	Pengguna berada di halaman grafik
Hasil Pengujian	Berhasil



Gambar 5.9 Hasil Pengujian Melihat Grafik

5.3 Evaluasi Pengujian

Berdasarkan hasil pengujian fungsionalitas dan akurasi sistem yang telah disampaikan, pada subbab ini akan dijelaskan mengenai evaluasi dari masing-masing pengujian.

5.3.1 Evaluasi Pengujian *Heuristic*

Setelah menyelesaikan evaluasi *heuristic* pada aplikasi ClearRoute, diidentifikasi sembilan area masalah yang menyalahi prinsip *usability*. Delapan permasalahan dapat dilihat di Tabel 5.14 dan Tabel 5.15 berikut.

Tabel 5.14 Daftar Masalah Dari Pengujian

No	Masalah
1	Pada halaman peta, kurang panduan pencarian
2	Pada halaman grafik, <i>icon</i> dan tulisan kurang tepat
3	Pada halaman berita, batas <i>post</i> terlalu luas
4	Pada halaman berita, <i>icon</i> tambah berita sulit ditemukan
5	Pada halaman cuaca, bisa ditambahkan <i>icon</i> bantuan
6	Pada halaman cuaca, tata letak informasi kurang keatas
7	Pada halaman berita, tata letak <i>icon</i> kurang <i>clean design</i>
8	Pada halaman berita, kata-kata kurang tepat

Tabel 5.15 Analisa Heuristic Permasalahan

No	Tingkat Gangguan	Tingkat Perbaikan	Nomor Heuristic
1	1	1	#2, #9
2	0	0	#1, #3
3	3	2	#4
4	3	2	#1, #4
5	2	1	#1, #2, #9
6	0	0	#1
7	1	0	#1
8	0	0	#3

Untuk lebih detail akan dijelaskan pada uraian dibawah.

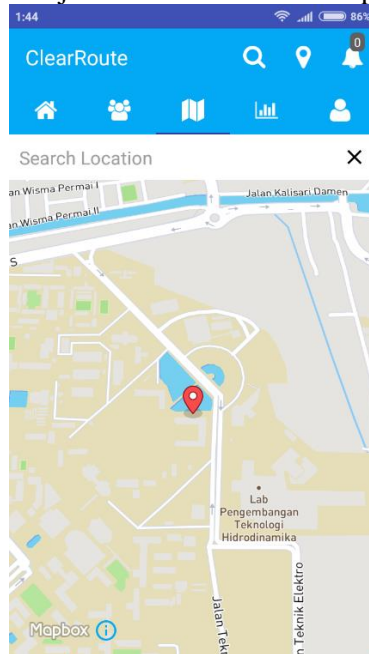
1. Kurang panduan pencarian

- Masalah

Pada saat pengguna mengakses halaman peta, bagi pengguna awam akan merasa seringkali tidak tahu harus melakukan apa di halaman tersebut. Struktur penempatan kolom pencarian dirasa kurang dapat mudah dikenali pengguna.

➤ **Bukti**

Kurangnya petunjuk penggunaan dan struktur penempatan kolom pencarian yang tidak biasanya menjadi faktor ketidaktahuan pengguna.



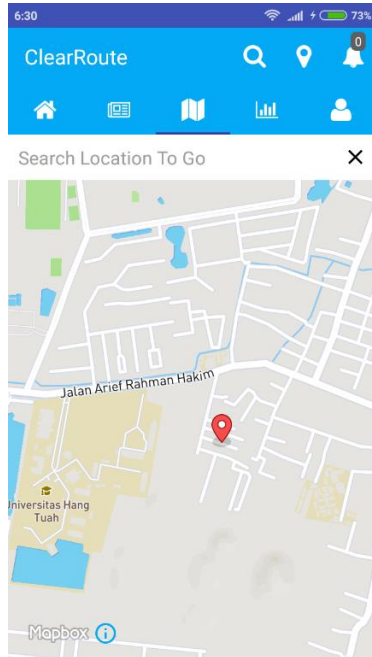
Gambar 5.10 Bukti Masalah 1

➤ **Rekomendasi**

Cukup ganti kata “*Search Location*” dengan kata yang lebih familiar seperti “*Go To*” atau kata lain yang lebih mudah dilihat dan dibaca.

Bisa juga diletakkan di *toolbar* untuk pencarian lokasi dikarenakan pada umumnya pencarian ada pada *toolbar*

➤ Revisi



Gambar 5.11 Revisi Masalah 1

2. *Icon* dan tulisan kurang tepat

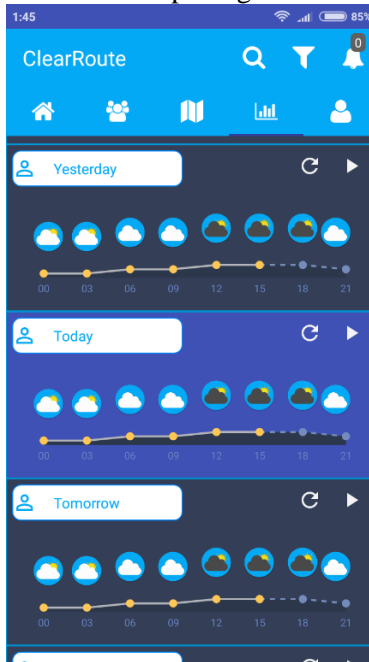
➤ Masalah

Bentuk *icon* yang tidak representatif dengan kata-kata yang tidak familiar atau disingkat menjadi masalah pada pengguna yang tidak familiar dengan bahasa tersebut. Dari kata-kata dan *icon* yang tidak representatif ini dikhawatirkan informasi yang ada tidak tersampaikan dengan baik.

➤ Bukti

Bentuk *icon* manusia yang tidak merepresentasikan cuaca maupun grafik dinilai

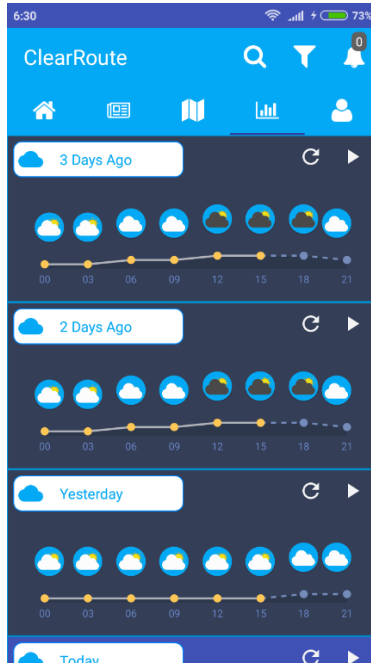
tidak tepat. Kata-kata seperti “H-3”, “H+2” juga dirasa tidak tepat digunakan.



Gambar 5.12 Bukti Masalah 2

- **Rekomendasi**
Icon yang digunakan akan lebih baik menggunakan *icon* cuaca atau logo ClearRoute itu sendiri. Sedangkan kata-kata yang tepat untuk menggantikan “H – 3” bisa dengan kata “3 days ago” dan sebagainya.

➤ Revisi



Gambar 5.13 Revisi Masalah 2

3. Batas *post* terlalu luas

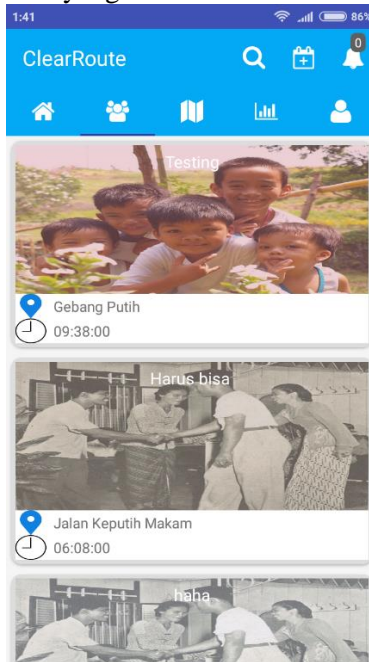
➤ Masalah

Postingan yang begitu banyak tentunya membutuhkan pengambilan data yang besar dan memakan waktu yang banyak. Hal ini dapat mempengaruhi kenyamanan pengguna dalam mendapatkan berita. Terlebih lagi aplikasi ini sangat menggantungkan diri pada koneksi yang stabil agar bisa dengan nyaman digunakan.

➤ Bukti

Kumpulan berita yang tidak tertata dalam hal penampilan data yang begitu banyak. Hal ini

membuat begitu sia-sianya waktu pengambilan data yang dilakukan.



Gambar 5.14 Bukti Masalah 3

- Rekomendasi
Untuk penampilan kumpulan berita akan lebih baik jika diurutkan hingga 3 hari terakhir saja sehingga tidak begitu banyak waktu dan data yang dibutuhkan.
- Revisi

```
$sql = 'select * from event where timestamp > NOW() - INTERVAL 3 DAY order by timestamp DESC';
$result = $con->query($sql);
while($row = mysqli_fetch_assoc($result)){
    $emparray[] = $row;
}
echo json_encode($emparray);
```

Kode Sumber 5.1 Revisi Masalah 3

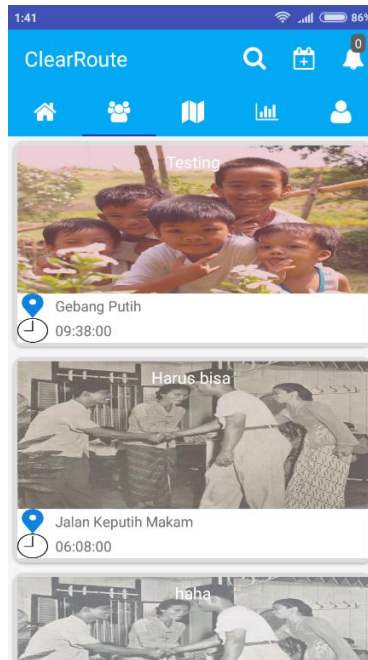
4. *Icon* tambah berita sulit ditemukan

➤ Masalah

Pada saat melakukan penambahan berita cukup menyita waktu dikarenakan pengguna tidak familiar dengan *icon* dan letak *icon* tersebut serta penempatan yang cukup susah dijangkau jari tangan.

➤ Bukti

Terbukti pada saat pengujian fungsionalitas yang mana mayoritas penguji lebih kesusahan dalam menemukan letak *icon* untuk menambah berita.



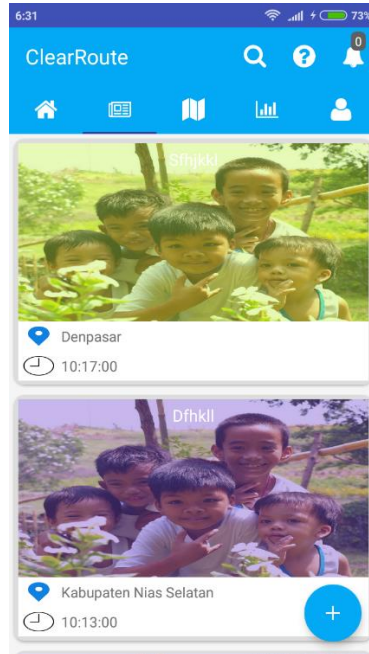
Gambar 5.15 Bukti Masalah 4

➤ Rekomendasi

Akan lebih baik jika digunakan *bubble* pada pojok kanan bawah layar ponsel seperti pada Path

agar lebih memudahkan pengguna dalam mengakses fitur penambahan berita.

➤ Revisi



Gambar 5.16 Revisi Masalah 4

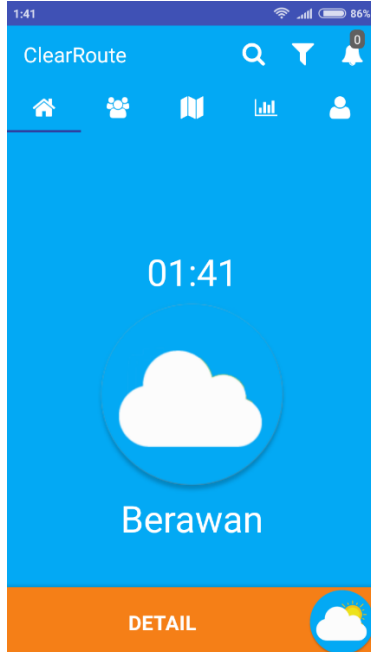
5. Bisa ditambahkan *icon* bantuan

➤ Masalah

Pada halaman cuaca adanya satu *icon* pada *toolbar* yang masih belum ada kegunaan yang signifikan. Selain itu, penggunaan aplikasi ini belum ada panduan satupun yang menjelaskan mengenai setiap halaman dan tombol *icon* pada aplikasi agar pengguna awam dapat memahami aplikasi lebih awal.

➤ **Bukti**

Kurangnya panduan membuat pekerjaan pengguna bisa lebih lama dari biasanya. Meskipun hal ini dapat dipercepat dengan semakin terbiasanya pengguna dengan aplikasi tersebut.



Gambar 5.17 Bukti Masalah 5

➤ **Rekomendasi**

Akan lebih baik jika *icon Filter* pada *toolbar* diganti dengan *icon Help* untuk mempermudah penggunaan aplikasi ini.

➤ Revisi



Gambar 5.18 Revisi Masalah 5

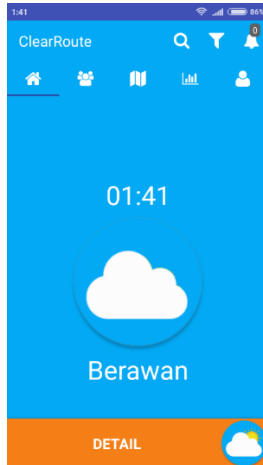
6. Tata letak informasi kurang keatas

➤ Masalah

Tata letak informasi yang kurang estetik mungkin bagi sebagian pihak dianggap bukan sesuatu yang begitu penting. Tapi sebagian pihak yang lain, tata letak yang kurang pas bisa memberikan dampak psikologis pada manusia terutama manusia dengan sifat perfeksionis.

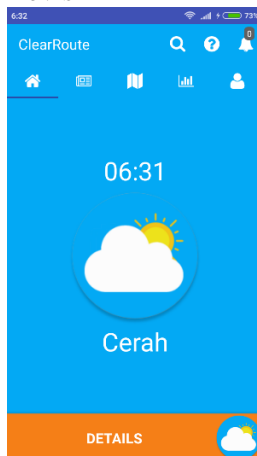
➤ Bukti

Tata letak informasi waktu hingga kondisi cuaca yang kurang sentris di tengah layar.



Gambar 5.19 Bukti Masalah 6

- Rekomendasi
Akan lebih baik jika letak informasi tersebut diletakkan lebih keatas atau lebih sentris lagi.
- Revisi



Gambar 5.20 Revisi Masalah 6

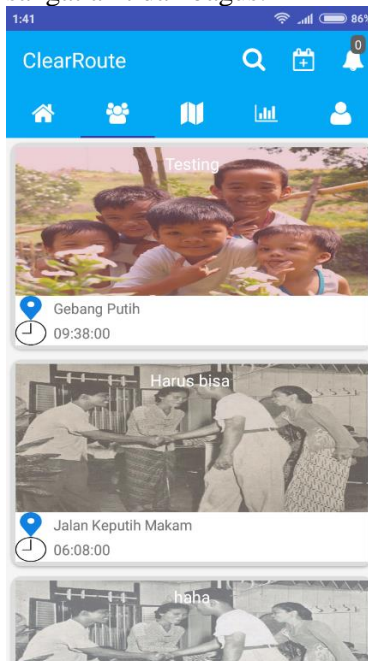
7. Tata letak *icon* kurang *clean design*

➤ Masalah

Tata letak yang tidak bagus dapat membuat misinterpretasi oleh pengguna mengenai fungsi *icon* tersebut. Hal ini sering terjadi pada letak *icon-icon* yang terlalu dekat atau tumpang tindih dan jari manusia yang begitu besar bisa menyebabkan salah pemanggilan tombol atau fungsi.

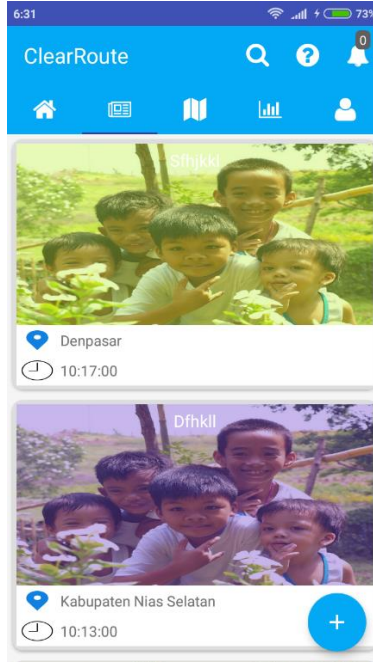
➤ Bukti

Tata letak pada daftar berita terutama pada *icon* waktu dan tempat yang cukup berdekatan sangatlah tidak bagus.



Gambar 5.21 Bukti Masalah 7

- Rekomendasi
Akan lebih baik jika *icon-icon* tersebut diatur ulang jarak dan letaknya. Meskipun satu kelompok informasi detail berita, *icon* yang terlalu dekat juga tidak bagus.
- Revisi

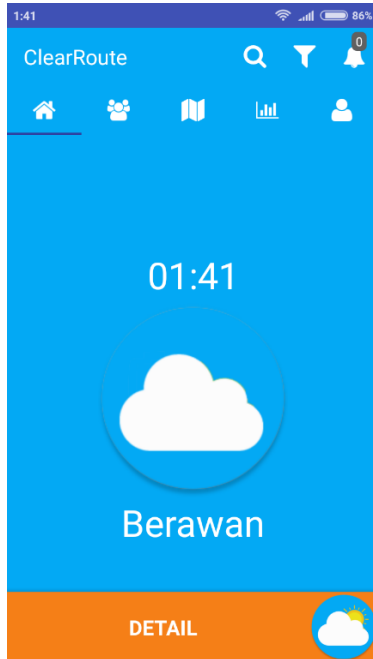


Gambar 5.22 Revisi Masalah 7

8. Kata-kata kurang tepat

- Masalah
Penggunaan kata-kata yang tidak tepat dapat membuat perbedaan interpretasi oleh pengguna. *Typo* ataupun *grammar nazi*-pun cukup berpengaruh pada penggunaan aplikasi yang dilakukan oleh beragam pengguna nantinya.

- **Bukti**
Penggunaan kata “Detail” yang menunjukkan data lebih dari satu buah dirasa kurang cocok.



Gambar 5.23 Bukti Masalah 8

- **Rekomendasi**
Akan lebih baik jika kata “Detail” diganti dengan kata “Details”

➤ Revisi



Gambar 5.24 Revisi Masalah 8

5.3.2 Evaluasi Pengujian Fungsionalitas

Berdasarkan hasil pelaksanaan pengujian terdapat beberapa hal yang bisa dirangkumkan dalam beberapa tabel untuk dianalisa.

Pada Tabel 5.16 dapat dilihat mengenai seberapa berhasil sistem yang dibangun pada aplikasi yang diujicobakan pada sepuluh orang berbeda.

Tabel 5.16 Hasil Pengujian Fungsional

No	ID	Kebutuhan Fungsional	Terpenuhi
1	UJ-001	Pengujian mendaftar akun	10
2	UJ-002	Pengujian melakukan login	10

3	UJ-003	Pengujian melihat profil	10
4	UJ-004	Pengujian mengubah profil	10
5	UJ-005	Pengujian melihat ramalan terkini	10
6	UJ-006	Pengujian melihat berita cuaca	10
7	UJ-007	Pengujian membuat berita cuaca	10
8	UJ-008	Pengujian melihat rekomendasi rute	10
9	UJ-009	Pengujian melihat grafik	10

Dari data pada Tabel 5.16 terlihat bahwa sistem secara penuh berhasil menjalankan fungsionalitasnya. Dari sepuluh orang penguji, tak satupun mengalami kegagalan fungsional.

Pada Tabel 5.17 dapat dilihat mengenai seberapa lama pengguna melakukan pengujian fungsionalitas aplikasi ini.

Tabel 5.17 Waktu Pengujian Fungsional

No	ID	Kebutuhan Fungsional	Rata-rata
1	UJ-001	Pengujian mendaftar akun	44,1 detik
2	UJ-002	Pengujian melakukan login	17,3 detik
3	UJ-003	Pengujian melihat profil	4,6 detik
4	UJ-004	Pengujian mengubah profil	21,5 detik
5	UJ-005	Pengujian melihat ramalan terkini	5,8 detik
6	UJ-006	Pengujian melihat berita cuaca	6,3 detik
7	UJ-007	Pengujian membuat berita cuaca	51,4 detik
8	UJ-008	Pengujian melihat rekomendasi rute	4,7 detik
9	UJ-009	Pengujian melihat grafik	3,5 detik

Dari data pada Tabel 5.17 terlihat keberagaman waktu yang dibutuhkan pengguna untuk menyelesaikan setiap fungsionalitas yang ada. Keberagaman tersebut memiliki beberapa faktor diantaranya ialah :

1. Adanya proses memasukkan data yang membutuhkan waktu lebih lama.
2. Pencarian lokasi fungsionalitas yang kurang memadai dikarenakan *icon* yang digunakan tidak familiar dengan pengguna.
3. Kualitas koneksi internet pada *smartphone* yang digunakan mempengaruhi proses pengambilan dan pengiriman data.
4. Proses menggeser dengan jari lebih membutuhkan sedikit waktu daripada menekan tombol-tombol. Hal ini terjadi dikarenakan sensitifitas *smartphone* pada saat ditekan lebih kecil dibandingkan dengan digeser.

Pada Tabel 5.18 dapat dilihat mengenai seberapa efektifnya unsur-unsur antarmuka yang digunakan terhadap kenyamanan pengguna.

Tabel 5.18 Hasil Pengujian Antarmuka

No	Parameter	Nilai
1	<i>Icon</i>	3,4
2	<i>Font</i>	4,5
3	<i>Layout</i>	4,5
4	Bahasa	3,6
5	Tata Letak	3,8

Dari Tabel 5.18 dapat diketahui bahwa aplikasi sudah memenuhi standar kenyamanan pengguna jika kita menggunakan nilai 3 sebagai standar kenyamanan pengguna. Akan tetapi, nilai tersebut juga diikuti dengan beberapa masukan dari pengguna diantaranya ialah :

3. Beberapa *icon* tidak mempresentasikan maksudnya. Seperti *icon* berita dan rute
4. Bahasa yang digunakan harus konsisten. Seperti, kata pada ubah profil menggunakan kata “kembali”
5. Pewarnaan *font* disesuaikan dengan *background*
6. Ada kata-kata yang kurang dimengerti
7. Menu *logout* harus *klik* profil dulu
8. *Font* bagus dan besar
9. Tidak ada rasio *layout* yang jelek
10. Kata-katanya bagus dan mudah dipahami namun tidak banyak adanya penjelasan
11. Susah menemukan tambah berita dan grafik dirasa terbalik

Setelah melihat nilai dan pernyataan yang diberikan pengguna, aplikasi sudah memberikan kenyamanan akan tetapi akan lebih baik aplikasi diperbaharui kembali dengan memperbaiki beberapa hal dalam desain yang dirasa kurang memberikan kenyamanan.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

6.1 Kesimpulan

Dalam proses pengerjaan Tugas Akhir dari tahap pendahuluan, kajian pustaka, analisis, perancangan, implementasi dan pengujian metode *Gestalt Principles* pada aplikasi android ClearRoute diperoleh kesimpulan sebagai berikut.

- a. Pembangunan aplikasi yang didasarkan pada sembilan kasus penggunaan utama dapat berfungsi seluruhnya dengan melakukan pengujian *blackbox*.
- b. Dari tiga fungsionalitas yang memanfaatkan data lokasi dan kondisi cuaca dari BMKG didapatkan waktu penggunaan 3,5 hingga 5,8 detik dalam pengujian *blackbox*.
- c. Dari delapan masalah dalam pengujian *heuristic*, mayoritas gangguan berada pada tingkat 0 atau tidak mengganggu penggunaan dan mayoritas perbaikan pada tingkat 0 atau sangat mudah untuk diselesaikan. Meskipun demikian, terdapat gangguan tingkat 3 atau gangguan penggunaan besar dan perbaikan tingkat 2 atau sulit untuk diselesaikan.
- d. Dari pengujian fungsionalitas, didapatkan rating nilai 3,4 hingga 4,5 dari skala 5 untuk antarmuka aplikasi.

- e. Setiap antarmuka aplikasi ClearRoute sudah menerapkan prinsip *Gestalt* berdasarkan pengujian *heuristic*. Prinsip *Gestalt* yang digunakan yaitu *proximity*, *figure/ground*, dan *common fate*.

6.2 Saran

Berikut ini merupakan beberapa saran mengenai pengembangan lebih lanjut aplikasi ClearRoute dengan penerapan *Gestalt Principles* berdasarkan hasil rancangan, implementasi dan uji coba yang telah dilakukan.

- a. Lebih banyak melakukan *alpha* dan *beta testing* sebelum dilakukannya pengujian aplikasi ke khalayak umum.
- b. Ujicoba dapat dilakukan kepada seluruh versi OS, dari Jelly Bean hingga Nougat dan beragam resolusi ponsel untuk mendapatkan hasil ujicoba yang lebih akurat.
- c. Jika memungkinkan untuk tetap memperhatikan 3 prinsip lain pada *Gestalt Principles* untuk mendapatkan desain aplikasi yang lebih baik.
- d. Untuk pengunggahan gambar dapat menggunakan Multipart dikarenakan efisiensi ukuran dan waktu pengunggahan.

DAFTAR PUSTAKA

- [1] “Android,” *Android*. [Online]. Available: <https://www.android.com/>. [Accessed: 13-Jul-2017].
- [2] “Free Community-based Mapping, Traffic & Navigation App.” [Online]. Available: <https://www.waze.com/>. [Accessed: 13-Jul-2017].
- [3] Albertrahmat |, “Mengenal Google Maps,” *Laboratory of Geographic Information System*, 17-Sep-2015. .
- [4] “Mobile Apps – BMKG.” [Online]. Available: <http://apps.bmkg.go.id/>. [Accessed: 13-Jul-2017].
- [5] BMKG 2017 and B. M. Geofisika Klimatologi dan, “Tugas dan Fungsi | BMKG,” *BMKG / Badan Meteorologi, Klimatologi, dan Geofisika*. [Online]. Available: ?p=tugas-fungsi&lang=ID. [Accessed: 13-Jul-2017].
- [6] S. Kumar, M. A. Qadeer, and A. Gupta, *Location Based Services using Android (LBSOID)*. Bangalore, India: IEEE, 2009.
- [7] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, Third. Harlow: Pearson Prentice Hall, 2004.
- [8] D. A. Norman, *The Design of Everyday Things: Revised and Expanded Edition*. New York: Basic Books (AZ), 2013.
- [9] A. Marcus, *Mobile Persuasion Design*. Berkeley: Springer, 2014.
- [10] I. S. MacKenzie, *Human-Computer Interaction, An Empirical Research Perspective*. Waltham: Elsevier, 2013.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

8.1 Kode Sumber pada Aplikasi

```
Dexter.withActivity(this)
    .withPermissions(
        Manifest.permission.INTERNET,
        Manifest.permission.WRITE_EXTERNAL_STORAGE,
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.ACCESS_NETWORK_STATE
    ).withListener(new MultiplePermissionsListener() {
        @Override public void onPermissionsChecked(MultiplePermissionsReport report) {
            new Handler().postDelayed(() -> {
                Intent i = new Intent(SplashScreenActivity.this, LoginActivity.class);
                startActivity(i);
                finish();
            }, 1500);
        }

        @Override
        public void onPermissionRationaleShouldBeShown(List<com.karumi.dexter.listener.PermissionRequest> permissions,
            PermissionToken token) {
        }
    })
    .withErrorListener((error) -> {
        Log.e("Dexter", "There was an error: " + error.toString());
    })
    .check();
```

Kode Sumber 8.1 Fungsi *Dexter*

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_weight="1"
    android:padding="16dp"
    android:layout_gravity="center_vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/gradient_background"
    tools:context="com.azkaakh.clearroute.activity.LoginActivity">
    <ImageView...>

    <EditText...>
    <EditText...>

    <Button...>

    <LinearLayout...>
    <LinearLayout...>

</LinearLayout>

```

Kode Sumber 8.2 Pengaturan *Proximity* pada Halaman *Login*

```

private void login() {
    Call<Validate> log = service.login_validate(user, pass);
    log.enqueue(new Callback<Validate>() {
        @Override
        public void onResponse(Call<Validate> call, Response<Validate> response) {
            try {
                validate(response.body().validate);
            } catch (Exception e) {
                Snackbar.make(parentView, "Login bermasalah", Snackbar.LENGTH_LONG).show();
                e.printStackTrace();
            }
        }
        @Override
        public void onFailure(Call<Validate> call, Throwable t) {
            Snackbar.make(parentView, "Koneksi bermasalah", Snackbar.LENGTH_LONG).show();
        }
    });
}

```

Kode Sumber 8.3 Fungsi *Login*

```

public class LineCardOne extends CardController {
    private final LineChartView mChart;
    private final Context mContext;
    private final String[] mLabels = {"00", "03", "06", "09", "12", "15", "18", "21"};
    private final float[][] mValues = {{1f, 1f, 1f, 1f, 1f, 1f, 1f, 1f},
                                         {1f, 3f, 4f, 4f, 4f, 3f, 1f, 1f}};
    private Tooltip mTip;
    private Runnable mBaseAction;

    public LineCardOne(CardView card, Context context) {
        super(card);
        mContext = context;
        mChart = (LineChartView) card.findViewById(R.id.chart1);
    }

    @Override
    public void show(Runnable action) {...}
    @Override
    public void update() {...}
    @Override
    public void dismiss(Runnable action) {...}
    private void showTooltip(int tips, int place) {...}
}

```

Kode Sumber 8.4 Fungsi Pengaturan Animasi pada Grafik

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_vertical"
    android:orientation="vertical"
    android:paddingLeft="16dp"
    android:paddingRight="16dp">

    <EditText...>
    <EditText...>
    <EditText...>
    <EditText...>
    <EditText...>

    <Button...>

    <LinearLayout...>
</LinearLayout>
```

Kode Sumber 8.5 Pengaturan *Proximity* pada Halaman *Signup*

```

<!-- MAIN CONTENT -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary"
    android:gravity="center"
    android:orientation="vertical">
    <TextView...>
    <ImageView...>
    <TextView...>
</LinearLayout>

<!-- SLIDING LAYOUT -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary"
    android:orientation="vertical"
    android:clickable="true"
    android:focusable="false"
    android:id="@+id/dragView">

    <LinearLayout...>
    <LinearLayout...>

</LinearLayout>

```

Kode Sumber 8.6 Pengaturan *Common Fate* dan *Proximity* pada Halaman Ramalan

```

mLayout = (SlidingUpPanelLayout) layout.findViewById(R.id.sliding_layout);
mLayout.addPanelSlideListener(new SlidingUpPanelLayout.PanelSlideListener() {
    @Override
    public void onPanelSlide(View panel, float slideOffset) {
        Log.i(TAG, "onPanelSlide, offset " + slideOffset);
    }

    @Override
    public void onPanelStateChanged(View panel, SlidingUpPanelLayout.PanelState previousState,
        SlidingUpPanelLayout.PanelState newState) {
        Log.i(TAG, "onPanelStateChanged " + newState);
    }
});
mLayout.setFadeOnClickListener((view) -> {
    mLayout.setPanelState(SlidingUpPanelLayout.PanelState.COLLAPSED);
});

```

Kode Sumber 8.7 Fungsi *SlidingPanel*

```

mViewPager = (ViewPager) findViewById(R.id.container);
setupViewPager(mViewPager);
tabLayout = (TabLayout) findViewById(R.id.tabs);
tabLayout.setupWithViewPager(mViewPager);
setupTabIcons();
//getkategori();
tabLayout.addTabSelectedListener(new TabLayout.OnTabSelectedListener() { [...] });

```

Kode Sumber 8.8 Fungsi *TabLayout*

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.beranda, menu);
    menu.findItem(R.id.search_menu).setIcon(
        new IconicsDrawable(this)
            .icon(FontAwesome.Icon.faw_search)
            .color(Color.WHITE)
            .actionBar()
    );
    menu.findItem(R.id.filter).setIcon(
        new IconicsDrawable(HomeActivity.this)
            .icon(FontAwesome.Icon.faw_filter)
            .color(Color.WHITE)
            .actionBar()
    );
    ActionItemBadge.update(this, menu.findItem(R.id.notification_menu), FontAwesome.Icon.faw_bell,
        ActionItemBadge.BadgeStyles.DARK_GREY, badgeCount);
    this.menu=menu;
    return super.onCreateOptionsMenu(menu);
}

```

Kode Sumber 8.9 Fungsi *ToolbarMenu*

```

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
View view = LayoutInflater.from(getActivity()).inflate(R.layout.profile_information,null);
final EditText etname,etemail,etphone;
etname = (EditText)view.findViewById(R.id.etName);
etemail = (EditText)view.findViewById(R.id.etEmail);
etphone = (EditText)view.findViewById(R.id.etPhone);

builder.setView(view);
builder.setCancelable(false);
builder.setPositiveButton("Update", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {

    }
}).setNegativeButton("Batal", (dialog, which) → { dialog.cancel(); });

//set the name
Call<User> log = service.getUser(Prefs.getString("username",""));
log.enqueue(new Callback<User>() {
    @Override
    public void onResponse(Call<User> call, Response<User> response) {...}
    @Override
    public void onFailure(Call<User> call, Throwable t) {

    }
});

```

Kode Sumber 8.10 Fungsi *Update* Profil

8.2 Hasil Pengujian dan Kuisioner

Nama : *Prandi Jaya S*

Pekerjaan : *mahasiswa*

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	46 detik
2	Melakukan Login	✓	20 detik
3	Melihat Profil	✓	4 detik
4	Mengubah Profil	✓	11 detik
5	Melihat Ramalan Cuaca	✓	10 detik
6	Melihat Berita Cuaca	✓	3 detik
7	Melihat Rekomendasi Rute	✓	6 detik
8	Melihat Grafik	✓	5 detik
9	Membuat Berita Cuaca	✓	95 detik

Daftar Pertanyaan:

1. Apakah icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

1 2 3 ④ 5

Komentar :

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

1 2 3 4 ⑤

Komentar :

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

1 2 3 4 ⑤

Komentar :

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

1 2 ③ 4 5

Komentar :

5. Apakah tata letak dari tampilan aplikasi sudah baik

1 2 3 ④ 5

Komentar :

Gambar 8.1 Objek Uji Coba 1

Nama : *M. Lutfi L.*

Pekerjaan : *Marketing*

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	33 detik
2	Melakukan Login	✓	12 detik
3	Melihat Profil	✓	3 detik
4	Mengubah Profil	✓	10 detik
5	Melihat Ramalan Cuaca	✓	4 detik
6	Melihat Berita Cuaca	✓	4 detik
7	Melihat Rekomendasi Rute	✓	3 detik
8	Melihat Grafik	✓	4 detik
9	Membuat Berita Cuaca	✓	62 detik

Daftar Pertanyaan:

1. Apakah icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

1 2 3 4 5

Komentar : *Cukup menarik tapi beberapa icon tidak representatif dan maknanya*

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

1 2 3 4 5

Komentar :

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

1 2 3 4 5

Komentar :

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

1 2 3 4 5

Komentar : *- Bahasa yang digunakan harus konsisten.
- Bahasa pada label profil pengguna menggunakan kata "kembali".*

5. Apakah tata letak dari tampilan aplikasi sudah baik

1 2 3 4 5

Komentar :

Gambar 8.2 Objek Uji Coba 2

Nama: Brian Rainer

Pekerjaan: Mahasiswa

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	35 detik
2	Melakukan Login	✓	18 detik
3	Melihat Profil	✓	11 detik
4	Mengubah Profil	✓	32 detik
5	Melihat Ramalan Cuaca	✓	13 detik
6	Melihat Berita Cuaca	✓	3 detik
7	Melihat Rekomendasi Rute	✓	4 detik
8	Melihat Grafik	✓	4 detik
9	Membuat Berita Cuaca	✓	63 detik

Daftar Pertanyaan:

1. Apakah Icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

Komentar: 1 2 3 ④ 5
halaman home/cuaca saat ini kurang jelas.

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

Komentar: 1 2 3 4 ⑤

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

Komentar: 1 2 3 ④ 5

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

Komentar: 1 2 ③ 4 5

5. Apakah tata letak dari tampilan aplikasi sudah baik

Komentar: 1 2 ③ 4 5

Gambar 8.3 Objek Uji Coba 3

Nama : Buchanuddin R

Pekerjaan : Mahasiswa

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	25 detik
2	Melakukan Login	✓	7 detik
3	Melihat Profil	✓	3 detik
4	Mengubah Profil	✓	76 detik
5	Melihat Ramalan Cuaca	✓	9 detik
6	Melihat Berita Cuaca	✓	2 detik
7	Melihat Rekomendasi Rute	✓	4 detik
8	Melihat Grafik	✓	3 detik
9	Membuat Berita Cuaca	✓	42 detik

Daftar Pertanyaan:

1. Apakah icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

Komentar: 1 2 3 4 5
Pute dan Berita kurang pos.

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

Komentar: 1 2 3 4 5
pewarnaan disesuaikan dengan background

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

Komentar: 1 2 3 4 5

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

Komentar: 1 2 3 4 5
Ada kata-kata yang kurang dimengerti

5. Apakah tata letak dari tampilan aplikasi sudah baik

Komentar: 1 2 3 4 5
menu logout harus ke profil dulu

Gambar 8.4 Objek Uji Coba 4

Nama : *Naufal Fathur M.*

Pekerjaan : *Marketing*

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	65 detik
2	Melakukan Login	✓	18 detik
3	Melihat Profil	✓	4 detik
4	Mengubah Profil	✓	12 detik
5	Melihat Ramalan Cuaca	✓	4 detik
6	Melihat Berita Cuaca	✓	18 detik
7	Melihat Rekomendasi Rute	✓	4 detik
8	Melihat Grafik	✓	2 detik
9	Membuat Berita Cuaca	✓	52 detik

Daftar Pertanyaan:

1. Apakah icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

Komentar : 1 2 3 4 5
icon melihat berita kurang cocok

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

Komentar : 1 2 3 4 5

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

Komentar : 1 2 3 4 5

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

Komentar : 1 2 3 4 5

5. Apakah tata letak dari tampilan aplikasi sudah baik

Komentar : 1 2 3 4 5

Gambar 8.5 Objek Uji Coba 5

Nama : Dhanar Praroga

Pekerjaan : Mahasiswa

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	46 Sec
2	Melakukan Login	✓	25 Sec
3	Melihat Profil	✓	0 : 03
4	Mengubah Profil	✓	0 : 07
5	Melihat Ramalan Cuaca	✓	0 : 07
6	Melihat Berita Cuaca	✓	0 : 05
7	Melihat Rekomendasi Rute	✓	0 : 02
8	Melihat Grafik	✓	0 : 03
9	Membuat Berita Cuaca	✓	0 : 25

Daftar Pertanyaan: Higher better

1. Apakah icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

Komentar: Icon maps & berita kering ¹ ² ³ ⁴ ⁵ Umum

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

Komentar: ¹ ² ³ ⁴ ⁵ Pewarnaan font pada berita buruk

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

Komentar: ¹ ² ³ ⁴ ⁵

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

Komentar: ¹ ² ³ ⁴ ⁵

5. Apakah tata letak dari tampilan aplikasi sudah baik

Komentar: ¹ ² ³ ⁴ ⁵

Gambar 8.6 Objek Uji Coba 6

Nama: Filtri

Pekerjaan: Mahasiswa

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	28 detik
2	Melakukan Login	✓	20 detik
3	Melihat Profil	✓	2 detik
4	Mengubah Profil	✓	10 detik
5	Melihat Ramalan Cuaca	✓	2 detik
6	Melihat Berita Cuaca	✓	4 detik
7	Melihat Rekomendasi Rute	✓	10 detik
8	Melihat Grafik	✓	2 detik
9	Membuat Berita Cuaca	✓	36 detik

Daftar Pertanyaan:

1. Apakah icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

Komentar: 1 2 3 4 5
Menu icon tidak sesuai

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

Komentar: 1 2 3 4 5
bagus dan besar

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

Komentar: 1 2 3 4 5
Tidak ada ruang yang jelek

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

Komentar: 1 2 3 4 5
bagus mudah dipahami namun tidak banyak penjelasan

5. Apakah tata letak dari tampilan aplikasi sudah baik

Komentar: 1 2 3 4 5
grafik terbalik, tidak dapat menampilkan tambah berita.

Gambar 8.7 Objek Uji Coba 7

Nama : William Budi J

Pekerjaan : Mahasiswa

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	42 detik
2	Melakukan Login	✓	21 detik
3	Melihat Profil	✓	3 detik
4	Mengubah Profil	✓	11 detik
5	Melihat Ramalan Cuaca	✓	2 detik
6	Melihat Berita Cuaca	✓	15 detik
7	Melihat Rekomendasi Rute	✓	3 detik
8	Melihat Grafik	✓	3 detik
9	Membuat Berita Cuaca	✓	53 detik

Daftar Pertanyaan:

1. Apakah icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

Komentar : ¹ icon ² petanya agak ³ membingungkan ⁴ 5

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

Komentar : ¹ ² ³ ⁴ ⁵

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

Komentar : ¹ ² ³ ⁴ ⁵

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

Komentar : ¹ ² ³ ⁴ ⁵

5. Apakah tata letak dari tampilan aplikasi sudah baik

Komentar : ¹ ² ³ ⁴ ⁵

Gambar 8.8 Objek Uji Coba 8

Nama : *Nizar*

Pekerjaan : *Mahasiswa*

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	44 detik
2	Melakukan Login	✓	12 detik
3	Melihat Profil	✓	4 detik
4	Mengubah Profil	✓	15 detik
5	Melihat Ramalan Cuaca	✓	3 detik
6	Melihat Berita Cuaca	✓	4 detik
7	Melihat Rekomendasi Rute	✓	4 detik
8	Melihat Grafik	✓	4 detik
9	Membuat Berita Cuaca	✓	47 detik

Daftar Pertanyaan:

1. Apakah Icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

1 2 3 4 5

Komentar : *icon melihat Berita kurang resmi*

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

1 2 3 4 5

Komentar :

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

1 2 3 4 5

Komentar :

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

1 2 3 4 5

Komentar :

5. Apakah tata letak dari tampilan aplikasi sudah baik

1 2 3 4 5

Komentar :

Gambar 8.9 Objek Uji Coba 9

Nama : *Muhammad Fathul Razi*

Pekerjaan : *Mahasiswa*

No	Fungsional	Terpenuhi	Waktu
1	Mendaftar Akun	✓	77 detik
2	Melakukan Login	✓	20 detik
3	Melihat Profil	✓	9 detik
4	Mengubah Profil	✓	31 detik
5	Melihat Ramalan Cuaca	✓	4 detik
6	Melihat Berita Cuaca	✓	5 detik
7	Melihat Rekomendasi Rute	✓	7 detik
8	Melihat Grafik	✓	5 detik
9	Membuat Berita Cuaca	✓	39 detik

Daftar Pertanyaan:

1. Apakah icon gambar pada aplikasi sudah cukup menarik dan membantu pengguna dalam menemukan menu yang diinginkan?

1 2 3 4 5

Komentar :

2. Apakah font yang digunakan sudah sesuai dan dapat dilihat dengan baik

1 2 3 4 5

Komentar :

3. Apakah layout yang digunakan sudah sesuai dengan ukuran layar

1 2 3 4 5

Komentar :

4. Apakah kata-kata yang digunakan pada halaman aplikasi mudah dipahami

1 2 3 4 5

Komentar :

5. Apakah tata letak dari tampilan aplikasi sudah baik

1 2 3 4 5

Komentar :

Gambar 8.10 Objek Uji Coba 10

Nama : Anjani
 Pekerjaan : Mahasiswa DKV

Daftar Evaluasi :

- Map Dikasih tulisan search to go
- Grafik "H-3" \Rightarrow "3 Days ago"
 "H+2" \Rightarrow "2 Days later"
- Grafik Iconnya bisa diganti matahari atau logo clearroute
- News Batas 3 hari sebelumnya
- News Icon buat nambah berita dibuat bubble bawah
- News Icon di list kurang Clean Design
- Home Bisa ditambahkan icon help
- Home Tulisan "Jam" hingga "Cerah" kurang keatas
- Home Tulisan "Detail" diganti: "Details"

Gambar 8.11 Hasil Evaluasi *Heuristic*

BIODATA PENULIS



Azkaa Khoiruddin, biasa dipanggil Azkaa, dilahirkan di kota Malang pada tanggal 28 Oktober 1994. Penulis adalah anak kedua dari tiga bersaudara dan dibesarkan di kota Malang, Jawa Timur. Penulis menempuh pendidikan di SDN Kebonsari 3 Malang, SMP Negeri 6 Malang dan SMA Negeri 3 Malang. Pada tahun 2013, penulis mengikuti SBMPTN dan diterima di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya yang terdaftar dengan NRP 5113100135. Di jurusan Teknik Informatika ini, penulis mengambil rumpun mata kuliah Algoritma dan Pemrograman. Selama di kuliah, penulis banyak belajar mengenai pemrograman Java, PHP, pemrograman perangkat bergerak. Penulis juga pernah menjadi asisten praktikum di beberapa mata kuliah seperti Dasar Pemrograman, Sistem Basis Data, Keamanan Informasi dan Jaringan, Aljabar Linier, dll. Selain itu, penulis aktif di beberapa organisasi di antaranya Staff Kesejahteraan Mahasiswa HMTK Berkarya, Staff Ahli Kewirausahaan HMTK Optimasi, Staff Kesekretariatan Schematics ITS 2014, Ketua Biro Web dan Kesekretariatan Schematics ITS 2015 dan Ketua GenBI Komisariat ITS 2016. Penulis dapat dihubungi melalui alamat surel **azkaa.khoiruddin.007@gmail.com**